MÉTODOS NUMÉRICOS CON SCILAB

Héctor Manuel Mora Escobar

hectormora@yahoo.com hmmorae@unal.edu.co
www.geocities.com/hectormora

July 31, 2009

ÍNDICE GENERAL

1

Preliminares

1.1 Notación

Sean a, b números reales, $a < b$. Los intervalos cerrados y abiertos son

$$[a, b] = \{x \in \mathbb{R} : a \leq x \leq b\},$$
$$]a, b[= \{x \in \mathbb{R} : a < x < b\}.$$

También es usual denotar el intervalo abierto por (a, b) pero puede confundirse con la pareja ordenada (a, b).

$C[a, b]$ es el conjunto de funciones continuas en el intervalo $[a, b]$; $C^n[a, b]$ es el conjunto de funciones con n derivadas continuas sobre $[a, b]$ (con esta notación $C[a, b] = C^0[a, b]$). Algunas veces, por brevedad, se dice que f es de clase C^n.

De manera análoga, $C^\infty[a, b]$ es el conjunto de funciones que se pueden derivar tantas veces como se desee. Ejemplos de estas funciones son: $f(x) = e^x$, $g(x) = 3x^5 - 8x^2 + 12$, $h(x) = \text{sen}(2x)$.

$\mathcal{P}_n$ es el conjunto de todos los polinomios de grado menor o igual a n.

$I(c, d)$ es el intervalo cerrado que más pequeño contiene a c y a d. Por ejemplo $I(3, 5) = [3, 5]$, $I(2, 1.8) = [1.8, 2]$.

1.2 Repaso de algunos conceptos de cálculo

En lo que sigue, mientras no se diga lo contrario, se considera una función $f : \mathbb{R} \to \mathbb{R}$ y c un número real.

Se dice que el *límite* de f cuando x tiende a c es $L \in \mathbb{R}$, denotado

$$\lim_{x \to c} f(x) = L\,,$$

si dado $\varepsilon > 0$ existe $\delta > 0$ tal que

$$\text{si } 0 < |x - c| \leq \delta, \quad \text{entonces} \quad |f(x) - L| \leq \varepsilon.$$

La función f es *continua* en c si $\lim_{x \to c} f(x)$ existe y

$$\lim_{x \to c} f(x) = f(c)\,.$$

Se dice que f es continua en el intervalo $[a, b]$ si es continua en todos los puntos de $[a, b]$.

Se dice que f es *derivable* en c, si existe el límite

$$\lim_{h \to 0} \frac{f(c+h) - f(c)}{h} = \lim_{\xi \to c} \frac{f(\xi) - f(c)}{\xi - c}\,.$$

Si f es derivable en c, entonces ese límite es la derivada de f en c y se denota

$$f'(c) = \lim_{h \to 0} \frac{f(c+h) - f(c)}{h} = \lim_{\xi \to c} \frac{f(\xi) - f(c)}{\xi - c}\,.$$

Teorema 1.1. ***Teorema de valores extremos****. Sea f continua en el intervalo $[a, b]$ (recordemos que se puede denotar $f \in C[a, b]$), entonces existe por lo menos un $\bar{x} \in [a, b]$ tal que*

$$f(\bar{x}) \leq f(x) \text{ para todo } x \in [a, b]\,.$$

Este punto $\bar{x}$ se llama minimizador absoluto o global o punto de mínimo global de f en $[a, b]$. De manera análoga, existe por lo menos un punto $\tilde{x}$, maximizador global o punto de máximo global, tal que

$$f(\tilde{x}) \geq f(x) \text{ para todo } x \in [a, b]\,.$$

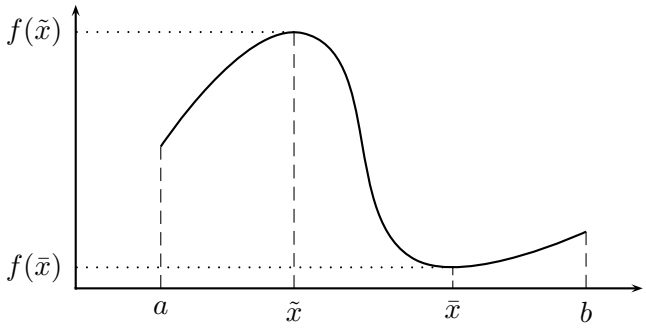


Figura 1.1: Teorema de valores extremos

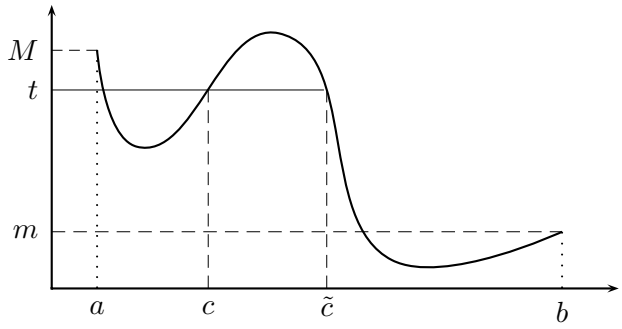


Figura 1.2: Teorema del valor intermedio

Teorema 1.2. ***Teorema del valor intermedio***. *Sea f continua en $[a, b]$, $m = \min\{f(a), f(b)\}$, $M = \max\{f(a), f(b)\}$. Si t es un valor intermedio, $m \leq t \leq M$, entonces existe por lo menos un $c \in [a, b]$ tal que*

$$f(c) = t.$$

Teorema 1.3. ***Teorema de Rolle***. *Si f es una función continua en $[a, b]$, derivable en $]a, b[$ y $f(a) = f(b)$, entonces existe $c \in]a, b[$ tal que*

$$f'(c) = 0.$$

Teorema 1.4. ***Teorema del valor medio***. *Si f es una función continua en $[a, b]$ y derivable en $]a, b[$, entonces existe $c \in]a, b[$ tal que*

$$f'(c) = \frac{f(b) - f(a)}{b - a}$$

.

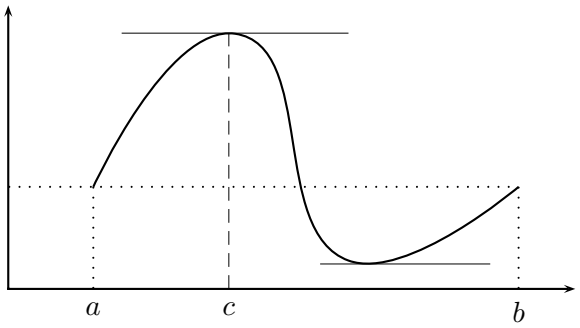


Figura 1.3: Teorema de Rolle

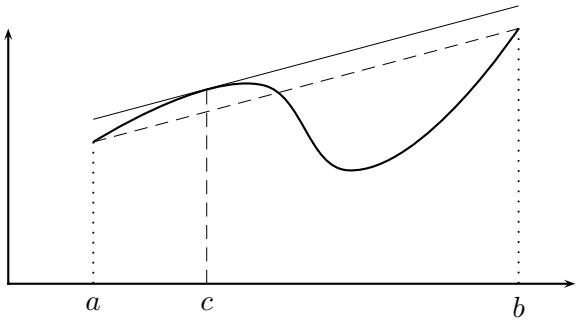


Figura 1.4: Teorema del valor medio

1.3 Sucesiones

Una sucesión es simplemente una función que va del conjunto de los números naturales en los reales:

$$\begin{aligned} u :&\mathbb{N} \to \mathbb{R} \\ &n \mapsto u_n \end{aligned}$$

Algunas veces las sucesiones están definidas para los naturales positivos (no para el 0). Como se observa, habitualmente se escribe u_n en lugar de $u(n)$. Es frecuente denotar la sucesión $\{u_n\}_{n\in\mathbb{N}}$ o $\{u_n\}_{n=0}^{\infty}$ o, si no hay confusión, de manera aún más simple, $\{u_n\}$ o u_n.

Ejemplos de sucesiones: $u_n = 1/n^2$, $v_n = 5+\dfrac{(-1)^n}{n^3}$, $w_m = \dfrac{m^3-2m}{100m^2+20m}$.

Una sucesión se puede definir de manera recurrente a partir del primer término o de los primeros términos. Por ejemplo, la sucesión de números de Fibonacci (Leonardo de Pisa) se define por:

$$\begin{aligned} u_0 &= 0 \\ u_1 &= 1 \\ u_n &= u_{n-2} + u_{n-1}, \quad \text{para } n \geq 2. \end{aligned}$$

Así $u_0 = 0$, $u_1 = 1$, $u_2 = 1$, $u_3 = 2$, $u_4 = 3$, $u_5 = 5$, $u_6 = 8$, $u_7 = 13$, , ...

Se dice que la sucesión x_n *converge* al número L, o que L es el límite de la sucesión, si dado cualquier $\varepsilon > 0$ (generalmente pequeño), existe un natural N tal que

$$|x_n - L| \leq \varepsilon \quad \text{para } n > N.$$

Es usual escribir

$$\lim_{n\to\infty} x_n = L$$

o

$$x_n \underset{n\to\infty}{\longrightarrow} L$$

o simplemente, si no hay confusión,

$$x_n \longrightarrow L$$

Ejemplo 1.1. Sea

$$x_n = 5 + \frac{1}{n^2} \ .$$

Veamos que el límite es 5. Si $\varepsilon = 0.01$, se requiere que

$$\begin{aligned} \left| 5 + \frac{1}{n^2} - 5 \right| &\leq 0.01 \\ \frac{1}{n^2} &\leq 0.01 \\ \frac{1}{0.01} &\leq n^2 \\ 100 &\leq n^2 \\ 10 &\leq n. \end{aligned}$$

Es decir para $\varepsilon = 0.01$ basta con tomar $N \geq 10$. En general para un ε cualquiera, basta con tomar $N \geq \sqrt{\frac{1}{\varepsilon}}$. $\diamond$

Se dice que la sucesión x_n tiende a $+\infty$ y se escribe

$$\lim_{n\to\infty} x_n = +\infty$$

o simplemente

$$x_n \longrightarrow +\infty$$

si dado cualquier real $M > 0$ (generalmenet grande), existe un natural N tal que

$$x_n > M \quad \text{para } n > N.$$

En este caso, la sucesión no es convergente pero, como se observa, se utiliza la misma notación. De manera análoga se define y denota cuando la sucesión tiende a $-\infty$.

La sucesión geométrica a^n converge o diverge dependiendo de a:

$$\begin{aligned}
&\lim_{n\to\infty} a^n = 0 && \text{si } |a| < 1,\\
&\lim_{n\to\infty} a^n = 1 && \text{si } a = 1,\\
&\lim_{n\to\infty} a^n = +\infty && \text{si } a > 1,\\
&\lim_{n\to\infty} a^n \text{ no existe} && \text{si } a \leq -1,
\end{aligned}$$

1.4 Polinomio de Taylor

Sea la función $f : \mathbb{R} \to \mathbb{R}$ continua y derivable cuantas veces sea necesario y sea $\bar{x}$ un valor fijo.

Se desea encontrar $p \in \mathcal{P}_1$ tal que

$$\begin{aligned}
p(\bar{x}) &= f(\bar{x}) \quad \text{y}\\
p'(\bar{x}) &= f'(\bar{x}).
\end{aligned}$$

Este polinomio es exactamente

$$p(x) = f(\bar{x}) + f'(\bar{x})(x - \bar{x}).$$

Ahora se desea encontar $p \in \mathcal{P}_2$ tal que

$$\begin{aligned}
p(\bar{x}) &= f(\bar{x}),\\
p'(\bar{x}) &= f'(\bar{x}),\\
p''(\bar{x}) &= f''(\bar{x}).
\end{aligned}$$

Entonces

$$p(x) = f(\bar{x}) + f'(\bar{x})(x - \bar{x}) + \frac{f''(\bar{x})}{2}(x - \bar{x})^2.$$

De manera general, sea $p \in \mathcal{P}_n$ tal que

$$\begin{aligned} p(\bar{x}) &= f(\bar{x}), \\ p'(\bar{x}) &= f'(\bar{x}), \\ p''(\bar{x}) &= f''(\bar{x}), \\ &\vdots \\ p^{(n)}(\bar{x}) &= f^{(n)}(\bar{x}). \end{aligned}$$

Este polinomio es

$$\begin{aligned} p(x) &= f(\bar{x}) + f'(\bar{x})(x-\bar{x}) + \frac{f''(\bar{x})}{2}(x-\bar{x})^2 + \cdots + \frac{f^{(n)}(\bar{x})}{n!}(x-\bar{x})^n \\ &= \sum_{k=0}^{n} \frac{f^{(k)}(\bar{x})}{k!}(x-\bar{x})^k \end{aligned} \tag{1.1}$$

llamado *polinomio de Taylor de orden n alrededor de* $\bar{x}$.

Teorema 1.5. *Sea* $f \in C^n[a,b]$, *tal que* $f^{(n+1)}$ *existe en* $[a,b]$ *y* $\bar{x} \in [a,b]$. *Entonces para todo* $x \in [a,b]$

$$f(x) = p_n(x) + R_n(x),$$

donde $p_n(x)$ *es el polinomio de Taylor y*

$$R_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-\bar{x})^{n+1} \tag{1.2}$$

es el residuo, con $\xi(x)$ *entre* $\bar{x}$ *y* x *(es decir,* $\xi(x) \in I(\bar{x}, x)$ *). Si* f *es de clase* C^∞, *entonces*

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(\bar{x})}{k!}(x-\bar{x})^k$$

La anterior expresión es el desarrollo en serie de Taylor de f *alrededor de* $\bar{x}$.

El teorema anterior no permite evaluar exactamente el residuo, pero si permite acotarlo:

$$|R_n(x)| \leq \frac{|x-\bar{x}|^{n+1}}{(n+1)!} \max_{t \in I(x,\bar{x})} \left| f^{(n+1)}(t) \right|$$

Ejemplo 1.2. Obtener la serie de Taylor de $f(x) = e^x$ alrededor de $\bar{x} = 0$.

$$\begin{aligned}
f'(x) &= e^x \\
f''(x) &= e^x \\
f^{(n)}(x) &= e^x \\
f(0) &= 1 \\
f'(0) &= 1 \\
f''(x) &= 1 \\
f^{(n)}(x) &= 1 \\
e^x &= 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots \\
e^x &= \sum_{n=0}^{\infty} \frac{x^n}{n!}
\end{aligned}$$

Ejemplo 1.3. Obtener la serie de Taylor de $f(x) = \operatorname{sen}(x)$ alrededor de $\bar{x} = 0$.

$$\begin{aligned}
f'(x) &= \cos(x) \\
f''(x) &= -\operatorname{sen}(x) \\
f'''(x) &= -\cos(x) \\
f^{(4)}(x) &= \operatorname{sen}(x) \\
f^{(5)}(x) &= \cos(x)
\end{aligned}$$

$$
\begin{aligned}
f(0) &= 0 \\
f'(0) &= 1 \\
f''(x) &= 0 \\
f'''(x) &= -1 \\
f^{(4)}(0) &= 0 \\
f^{(5)}(0) &= 1 \\
\operatorname{sen}(x) &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots
\end{aligned}
$$

Ejemplo 1.4. Obtener la serie de Taylor de $f(x) = \cos(x)$ alrededor de $\bar{x} = 0$.

$$
\begin{aligned}
f'(x) &= -\operatorname{sen}(x) \\
f''(x) &= -\cos(x) \\
f'''(x) &= \operatorname{sen}(x) \\
f^{(4)}(x) &= \cos(x) \\
f^{(5)}(x) &= -\operatorname{sen}(x) \\
f(0) &= 1 \\
f'(0) &= 0 \\
f''(x) &= -1 \\
f'''(x) &= 0 \\
f^{(4)}(0) &= 1 \\
f^{(5)}(0) &= 0 \\
\cos(x) &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots
\end{aligned}
$$

Ejemplo 1.5. Obtener el polinomio de Taylor de orden 2 de $\cos(x)$ alrededor de π, acotar el error para $x = 3$ y calcular el error.

$$
\begin{aligned}
p_2(x) &= \cos(\pi) - \operatorname{sen}(\pi)(x-\pi) - \frac{\cos(\pi)}{2}(x-\pi)^2 \\
p_2(x) &= -1 + \frac{1}{2}(x-\pi)^2
\end{aligned}
$$

$$
\begin{aligned}
|\text{error}| &\leq \frac{|3-\pi|^3}{6} \max_{t\in[3,\pi]} |\operatorname{sen}(t)| \\
|\text{error}| &\leq 0.0004731 \times \operatorname{sen}(3) \\
|\text{error}| &\leq 0.0004731 \times 0.1411 = 0.0000668 \\
|\text{error}| &\leq 0.0000668
\end{aligned}
$$

En este caso sencillo, se puede evaluar explícitamente el error:

$$
\begin{aligned}
|\text{error}| &= |\cos(3) - p_2(3)| \\
&= |-0.9899925 - -0.9899758| \\
&= 0.0000167 \qquad \diamond
\end{aligned}
$$

Algunas veces se expresa $x = \bar{x} + h$, entonces el polinomio de Taylor, el residuo y la serie de Taylor quedan:

$$p_n(\bar{x}+h) = \sum_{k=0}^{n} \frac{f^{(k)}(\bar{x})}{k!}\, h^k \tag{1.3}$$

$$R_n(\bar{x}+h) = \frac{f^{(n+1)}(\xi(h))}{(n+1)!}\, h^{n+1}, \quad \xi(h) \in I(0,h), \tag{1.4}$$

$$f(\bar{x}+h) = \sum_{k=0}^{\infty} \frac{f^{(k)}(\bar{x})}{k!}\, h^k. \tag{1.5}$$

1.5 Notación O grande

Algunas veces es útil comparar aproximadamente el comportamiento de dos funciones en las cercanías de 0.

Se dice que, cuando $x \to 0$,

$$f(x) = O(g(x))$$

si existen dos constantes positivas C y δ (pequeña) tales que

$$|f(x)| \leq C|g(x)| \quad \text{para} \quad |x| \leq \delta.$$

Ejemplo 1.6. Sea $f(x) = 4x^3+5x^6$. Recordemos que si $0 < y < 1$, entonces $y > y^2 > y^3 > \cdots$. Entonces, si $|x| < 1$,

$$\begin{aligned} |x^3| &\leq |x| \\ |4x^3| &\leq 4|x| \\ |x^6| &\leq |x| \\ |5x^6| &\leq 5|x| \\ |4x^3 + 5x^6| &\leq 9|x| \\ 4x^3 + 5x^6 &= O(x). \end{aligned}$$

Aunque lo anterior es cierto, es preferible buscar el mayor exponente posible. Mediante pasos semejante a los anteriores llegamos a

$$4x^3 + 5x^6 = O(x^3).$$

Obviamente no es cierto que $4x^3 + 5x^6 = O(x^4)$. $\diamond$

Según la notación O grande, el residuo para el polinomio de Taylor (1.4) se puede expresar

$$R_n(\bar{x} + h) = O(h^{n+1}).$$

1.6 Orden de convergencia

Sea $\{x_k\}$ una sucesión de números reales con límite L. Se dice que la convergencia tiene *orden de convergencia* $p \geq 1$, si p es el mayor valor tal que el siguiente límite existe.

$$\lim_{k\to\infty} \frac{|x_{k+1} - L|}{|x_k - L|^p} = \beta < \infty.$$

En este caso se dice que β es la tasa de convergencia. Cuando el orden es 1, se dice que la convergencia es lineal. La convergencia se llama *superlineal* si

$$\lim_{k\to\infty} \frac{|x_{k+1} - L|}{|x_k - L|} = 0.$$

Cuando el orden es 2, se dice que la convergencia es *cuadrática*.

Lo ideal es tener órdenes de convergencia altos con tasas pequeñas. Una convergencia lineal con tasa 1 es una convergencia muy lenta. Una convergencia cuadrática es muy buena, por ejemplo, el método de Newton que se verá más adelante.

Ejemplo 1.7. $x_k = \pi + \dfrac{1}{k}$. Esta sucesión converge a π. Veamos que pasa con $p = 1$.

$$\begin{aligned}
\lim_{k\to\infty} \frac{|x_{k+1} - L|}{|x_k - L|} &= \lim_{k\to\infty} \frac{|\pi + \dfrac{1}{k+1} - \pi|}{|\pi + \dfrac{1}{k} - \pi|} \\
&= \lim_{k\to\infty} \frac{\dfrac{1}{k+1}}{\dfrac{1}{k}} \\
&= \lim_{k\to\infty} \frac{k}{k+1} \\
&= 1.
\end{aligned}$$

Luego la sucesión tiene orden de convergencia por lo menos igual a 1. Veamos que pasa con $p > 1$. Se puede suponer que $p = 1 + \varepsilon$, con $\varepsilon > 0$.

$$\begin{aligned}
\lim_{k\to\infty} \frac{|x_{k+1} - L|}{|x_k - L|^{1+\varepsilon}} &= \lim_{k\to\infty} \frac{\dfrac{1}{k+1}}{\dfrac{1}{k^{1+\varepsilon}}} \\
&= \lim_{k\to\infty} \frac{k^{1+\varepsilon}}{1+k} \\
&= \lim_{k\to\infty} \frac{k\ k^{\varepsilon}}{1+k} \\
&= \left(\lim_{k\to\infty} \frac{k}{1+k}\right)\left(\lim_{k\to\infty} k^{\varepsilon}\right) \\
&= (1)(+\infty).
\end{aligned}$$

Entonces p no puede ser superior a 1 y podemos decir que la convergencia es lineal con tasa 1. $\diamond$

Ejemplo 1.8. $x_k = \dfrac{1}{2^k}$. Esta sucesión converge a 0. Directamente veamos que pasa con $p = 1 + \varepsilon$, con $\varepsilon \geq 0$.

$$\begin{aligned}
\lim_{k\to\infty} \frac{|x_{k+1}-L|}{|x_k-L|^{1+\varepsilon}} &= \lim_{k\to\infty} \frac{\dfrac{1}{2^{k+1}}}{\dfrac{1}{(2^k)^{1+\varepsilon}}} \\
&= \lim_{k\to\infty} \frac{(2^k)^{1+\varepsilon}}{2^{k+1}} \\
&= \lim_{k\to\infty} \frac{2^k\ 2^{k\varepsilon}}{2^{k+1}} \\
&= \left(\lim_{k\to\infty} \frac{2^k}{2^{k+1}}\right)\left(\lim_{k\to\infty} 2^{k\varepsilon}\right) \\
&= \left(\frac{1}{2}\right)\left(\lim_{k\to\infty} 2^{k\varepsilon}\right)
\end{aligned}$$

.

Si $p = 1$ ($\varepsilon = 0$) hay convergencia hacia $1/2$. Si $p > 1$ no hay convergencia. Entonces la sucesión tiene convergencia lineal con tasa $1/2$. $\diamond$

Ejemplo 1.9.

$$\begin{aligned}
x_1 &= \frac{69}{10} \\
x_n &= 6 + (x_{n-1} - 6)^2, \qquad n \geq 2.
\end{aligned}$$

Los primeros valores son los siguientes:

```
 1    6.900000000000000
 2    6.810000000000000
 3    6.656100000000000
 4    6.430467210000001
 5    6.185302018885185
 6    6.034336838202925
 7    6.001179018457774
 8    6.000001390084524
 9    6.000000000001933
10    6.000000000000000
```

Se puede mostrar que

$$\begin{aligned} x_n &= 6 + y_n\,, \quad n = 1, 2, ... \\ y_1 &= \frac{9}{10} \\ y_n &= y_{n-1}^2\,, \quad n = 2, 3, ... \\ y_n &= \left(\frac{9}{10}\right)^{2^{n-1}}\,, \quad n = 1, 2, ... \end{aligned}$$

Como $y_n \to 0$, entonces $x_n \to 6$.

$$\begin{aligned} \lim_{k\to\infty} \frac{|x_{k+1} - L|}{|x_k - L|^p} &= \lim_{k\to\infty} \frac{y_{k+1}}{y_k^p} \\ &= \lim_{k\to\infty} \frac{y_k^2}{y_k^p} \\ &= \lim_{k\to\infty} y_k^{2-p} \end{aligned}$$

Si $p = 1$ el límite es 0, es decir, la convergencia es por lo menos lineal y se puede afirmar que es superlineal. Si $p = 2$ el límite es 1, luego la convergencia es por lo menos cuadrática. Si $p = 2 + \varepsilon$, con $\varepsilon > 0$

$$\begin{aligned} \lim_{k\to\infty} \frac{|x_{k+1} - L|}{|x_k - L|^p} &= \lim_{k\to\infty} y_k^{2-(2+\varepsilon)} \\ &= \lim_{k\to\infty} \frac{1}{y_k^{\varepsilon}} \\ &= +\infty \end{aligned}$$

Luego la convergencia es cuadrática con tasa 1. $\diamond$

1.7 Números en un computador

Sea x un múmero real positivo. La representación decimal normalizada de x en un computador, con k cifras significativas es

$$\tilde{x} = 0.d_1 d_2 ... d_k \times 10^n$$

donde d_i es un entero en el intervalo $[0, 9]$ y $d_1 \geq 1$. El valor k, los valores mínimo y máximo permitidos para n dependen del computador, del sistema operativo o del lenguaje. Una manera aproximada de obtener estos valores en Scilab es la siguiente:

```
format(30)
x = 1/3
```

El resultado es

```
0.333333333333333148296
```

Únicamente hay **16** dígitos correctos, los demás son "basura" producida por Scilab para satisfacer el formato deseado. Esto nos indica que en Scilab, en la representación interna de un número, no hay más de 16 cifras significativas.

Relacionado con el concepto anterior, está el épsilon de la máquina, que se define así:

$$\varepsilon_{\text{maq}} = \min\{t > 0 : 1 + t \neq 1\}$$

La anterior definición usa los números utilizados en el computador. Este conjunto de números es finito y la definición tiene sentido. Obviamente si los valores t se tomaran en $\mathbb{R}$, el valor épsilon de la máquina estaría mal definido.

Una manera aproximada de obtener el épsilon de la máquina consiste en buscar, por ensayo y error, un valor x tal que $1 + x > 1$ y $1 + x/10 = 1$. La orden

```
x = 1.0e-10; x1 = 1+x; x2 = 1+x/10; (x1 > 1) & (x2 == 1)
```

produce `F` ("false"), en cambio

```
x = 1.0e-15; x1 = 1+x; x2 = 1+x/10; (x1 > 1) & (x2 == 1)
```

produce `T` ("true"). Esto nos indica que un valor aproximado es justamente 10^{-15}. Scilab tiene una valor predefinido

```
%eps = 2.220E-16
```

Para averiguar si un número positivo y pequeño es considerado como nulo, se puede ensayar con diferentes valores de la potencia de 10, por ejemplo:

```
x = 1.0e-20; x == 0.0
```

produce como resultado `F`, indicando que `x` no es nulo. Al ensayar

```
x = 1.0e-100; x == 0.0
```

el resultado de nuevo es `F`. Después de varios ensayos

```
x = 1.0e-323; x == 0.0
```

produce `F` y

```
x = 1.0e-324; x == 0.0
```

produce `T`, es decir, 10^{-324} **es considerado como nulo**.

Para evitar el ensayo y error se puede utilizar la siguiente secuencia de órdenes

```
x = 1;
while x/10 > 0.0
  x0 = x;
  x = x/10;
end
x_final = x0
```

El resultado obtenido es `9.881-323` . Obsérvese que `x` toma los valores 1, 1/10, 1/100, ... Sin embargo el resultado obtenido no es exactamente una potencia de 10.

Ahora queremos averiguar qué tan grandes pueden ser los números en Scilab. Así la orden

```
x = 1.0e308
```

muestra en la pantalla `1.000+308`, resultado esperado. La orden

```
x = 1.0e309
```

muestra en la pantalla `Inf` indicando que Scilab **considera** 10^{309} **como "infinito"** y no lo puede manejar adecuadamente.

1.8 Truncamiento y redondeo

Sea x un real (supuesto positivo por facilidad de presentación),

$$\tilde{x} = 0.d_1 d_2 ... d_k \times 10^n$$

su representación normalizada y t un entero positivo menor que k. El número obtenido por truncamiento con t cifras significativas es

$$\tilde{x}' = 0.d_1 d_2 ... d_t \times 10^n$$

Dicho de otra forma, se quitan los últimos $k - t$ dígitos. El redondeo con t cifras significativas se puede presentar de varias maneras equivalentes. Una de ellas es la siguiente,

$$\text{redondeo}(x,\, t) = \text{truncamiento}(\tilde{x} + 0.\underbrace{00\cdots0}_{t-1}5 \times 10^n)\,,\; t)$$

$$\begin{aligned}
\text{truncamiento}(1234.56789,\, 2) &= 1200 \\
\text{truncamiento}(1234.56789,\, 6) &= 1234.56 \\
\text{redondeo}(1234.56789,\, 2) &= 1200 \\
\text{redondeo}(1234.56789,\, 6) &= 1234.57
\end{aligned}$$

Una manera sencilla, que funciona cuando $d_t \leq 8$, es la siguiente: los primeros $t - 1$ dígitos son los mismos y el dígito en la posición t es

$$\delta_t = \begin{cases} d_t & \text{si} \quad d_{t+1} \leq 4 \\ d_t + 1 & \text{si} \quad d_{t+1} \geq 5. \end{cases}$$

Si $d_t = 9$ y $d_{t+1} \leq 4$, entonces $\delta_t = d_t$. Ahora bien, el caso especial se tiene si $d_t = 9$ y $d_{t+1} \geq 5$, entonces se suma 1 a $d_t = 9$, volviéndose 10 y se escribe $\delta_t = 0$, pero hay que agregar ("llevar") 1 al dígito d_{t-1}, etc.

1.9 Error absoluto y relativo

Si x es un número real y $\tilde{x}$ es una aproximación se definen el error absoluto (siempre no negativo) y el error relativo cuando $x \neq 0$, de la siguiente forma:

$$\begin{aligned}
\text{error absoluto} &= |x - \tilde{x}|\,, \\
\text{error relativo} &= \frac{|x - \tilde{x}|}{|x|}\,.
\end{aligned}$$

Ejemplo 1.10. Sean x y y números reales, $\tilde{x}$ el redondeo de x con $n = 5$ cifras significativas, $\tilde{y}$ el redondeo de y con n cifras significativas, $z = x - y$, $\tilde{z}$ el redondeo de $\tilde{x} - \tilde{y}$ con n cifras significativas, e_a el error absoluto entre z y $\tilde{z}$, e_r el error relativo.

x	y	$\tilde{x}$	$\tilde{y}$	z	$\tilde{z}$	e_a	e_r
1/7	2/3	0.14286	0.66667	$-11/21$	-0.52381	`4.8e-7`	`9.1e-7`
1/7	0.14284	0.14286	0.14284	0.00001714...	0.00002	`2.9e-6`	`1.7e-1`

En el segundo caso, el error relativo es grande, aproximadamente 17%. $\diamond$

Los principales casos en los que los errores pueden ser grandes, son:

1. Suma de cantidades de tamaños muy diferentes.
2. Resta de cantidades muy parecidas.
3. División por un número cercano a cero.

Estos casos, en cuanto sea posible, deben ser evitados y, si no es posible, los resultados deben ser interpretados de manera muy cuidadosa.

1.10 Errores lineal y exponencial

En los proceso numéricos, muy frecuentemente, es necesario realizar muchas operaciones aritméticas. Sea e_0 el error inicial en los datos o en la primera operación y e_n el error después de n operaciones. El error inicial incide en las operaciones siguientes y los errores, en la gran mayoría de los casos, van aumentando progresivamente. Usualmente se dice que los errores se propagan de dos maneras:

- Error lineal: $e_n \approx nce_0$
- Error exponencial: $e_n \approx c^n e_0$, con $c > 1$.

Es claro que un error exponencial (propagación exponencial del error) es muy peligroso y no es conveniente utilizar un algoritmo con esta clase de error. Con base en el tipo de error, se habla de *algoritmos estables* cuando el error es lineal y de *algoritmos inestables* cuando el error es exponencial.

Ejemplo 1.11. Consideremos la sucesión definida así (ver [KiC94]):

$$\begin{aligned} & x_0 = 1 \\ & x_1 = 1/3 \\ (*) \quad & x_n = \frac{13}{3}x_{n-1} - \frac{4}{3}x_{n-2}, \quad n \geq 2. \end{aligned}$$

Se puede demostrar que

$$(**) \quad x_n = \frac{1}{3^n}, \quad n = 0, 1, 2, ...$$

La siguiente tabla muestra los valores de $\bar{x}_n$ obtenidos en Scilab aplicando la fórmula explícita (**), $\tilde{x}_n$ obtenido por la fórmula de recurrencia (*) con todas las cifras que utiliza Scilab, $\tilde{x}'_n$ obtenido por la fórmula de recurrencia (*) pero trabajando con 8 cifras significativas y $\tilde{x}''_n$ obtenido por la fórmula de recurrencia (*) pero trabajando con 4 cifras significativas.

n	$\bar{x}_n$ (**)	$\tilde{x}_n$ (*)	$\tilde{x}'_n$ 8 cifras	$\tilde{x}''_n$ 4 cifras
0	1.00000000	1.00000000	1.00000000	1.00000000
1	0.33333333	0.33333333	0.33333333	0.33330000
2	0.11111111	0.11111111	0.11111110	0.11100000
3	0.03703704	0.03703704	0.03703700	0.03670000
4	0.01234568	0.01234568	0.01234554	0.01100000
5	0.00411523	0.00411523	0.00411468	-0.00126000
6	0.00137174	0.00137174	0.00136954	-0.02012000
7	0.00045725	0.00045725	0.00044843	-0.08550000
8	0.00015242	0.00015242	0.00011715	-0.34370000
9	0.00005081	0.00005081	-0.00009025	-1.37500000
10	0.00001694	0.00001694	-0.00054728	-5.50000000
11	0.00000565	0.00000564	-0.00225123	-22.0000000
12	0.00000188	0.00000188	-0.00902562	-88.0000000
13	0.00000063	0.00000063	-0.03610937	-352.000000
14	0.00000021	0.00000021	-0.14443977	-1408.00000
15	0.00000007	0.00000006	-0.57775985	-5632.00000
16	0.00000002	-0.00000003	-2.31103960	-22520.0000
17	0.00000001	-0.00000020	-9.24415860	-90070.0000
18	0.00000000	-0.00000085	-36.9766340	-360300.000
19	0.00000000	-0.00000340	-147.906540	-1441000.00
20	0.00000000	-0.00001361	-591.626160	-5764000.00
21	0.00000000	-0.00005445	-2366.50460	-23060000.0
25	0.00000000	-0.01393856	-605825.110	-5.904E+09

Se observa que la fórmula de recurrencia es un proceso inestable. La inestabilidad se nota más cuando hay menos cifras significativas. $\diamond$

1.11 Condicionamiento de un problema

Supongamos que un problema se puede resolver de manera exacta. Se dice que un problema es *bien condicionado* si al hacer cambios pequeños en los datos, se obtienen cambios pequeños en la solución. Un problema es *mal condicionado* si al hacer cambios pequeños en los datos, puede haber cambios grandes en la solución.

Cuando no hay un método exacto de solución, se dice que un problema es *mal condicionado* si, para todos los métodos utilizados, al hacer cambios pequeños en los datos, puede haber cambios grandes en la solución.

Ejemplo 1.12. Consideremos el sistema de ecuaciones $Ax = b$, donde

$$A = \begin{bmatrix} 10.01 & 10.00 \\ 10.00 & 9.99 \end{bmatrix}, \qquad b = \begin{bmatrix} 20.01 \\ 19.99 \end{bmatrix}.$$

La solución exacta de este problema es

$$x = [1 \quad 1]^{\mathrm{T}},$$

Consideremos ahora un sistema de ecuaciones muy parecido, únicamente hay cambios pequeños en b, $Ax' = b'$, donde

$$b' = \begin{bmatrix} 20.02 \\ 19.98 \end{bmatrix}.$$

La solución exacta de este problema es

$$x' = [-1998 \quad 2002]^{\mathrm{T}},$$

Este problema es mal condicionado, cambios pequeños en b produjeron cambios grandes en la solución. Más adelante se verá cómo determinar el buen o mal condicionamiento de un sistema de ecuaciones. $\diamond$

2

Solución de sistemas lineales

Uno de los problemas numéricos más frecuentes, o tal vez el más frecuente, consiste en resolver un sistema de ecuaciones de la forma

$$Ax = b \tag{2.1}$$

donde A es una matriz cuadrada, de tamaño $n \times n$, invertible. Esto quiere decir que el sistema tiene una única solución.

Se trata de resolver un sistema de ecuaciones de orden mucho mayor que 2. En la práctica se pueden encontrar sistemas de tamaño 20, 100, 1000 o mucho más grandes. Puesto que se trata de resolver el sistema con la ayuda de un computador, entonces las operaciones realizadas involucran errores de redondeo o truncamiento. La solución obtenida no es absolutamente exacta, pero se desea que la acumulación de los errores sea relativamente pequeña o casi despreciable.

2.1 En Scilab

Para resolver (2.1) es necesario haber definido una matriz cuadrada `a` y un vector columna `b`. La orden es simplemente

```
x = a\b
```

Por ejemplo

```
a = [ 2 3; 4 5], b = [-5; -7], x=a\b
```

da como resultado

```
 x  =

    2.
  - 3.
```

Una manera que también permite obtener la solución es `x = inv(a)*b`, pero es ineficiente en tiempo y de menor precisión.

Ejemplo 2.1. Las siguientes órdenes de Scilab

```
n = 500;

a = rand(n,n);
x = rand(n,1);
b = a*x;

tic()
x1 = a\b;
t_sol = toc();

tic()
x2 = inv(a)*b;
t_inv = toc();

error1 = norm(x1-x);
error2 = norm(x2-x);

printf('t_sol     = %f        t_inv     = %f\n', t_sol, t_inv)
printf('error_sol = %e    error_inv = %e\n', error1, error2)
```

producen un resultado análogo a

```
t_sol     = 0.622000        t_inv     = 1.737000
error_sol = 7.990870e-12    error_inv = 1.687945e-11
```

Estos resultados dependen del computador, del sistema operacional y aún en el mismo computador no son siempre iguales, pero sí parecidos. Las funciones `tic` y `toc` permiten obtener una medida del tiempo de un proceso. ◇

2.2 Notación

Sean A una matriz $m \times n$, con elementos a_{ij}, $i = 1, ...m$, $j = 1, ..., n$ y $x = (x_1, x_2, ..., x_n)$. Para denotar filas o columnas, o partes de ellas, se usará la notación de Matlab y Scilab.

parte de un vector: $x(5:7) = (x_5, x_6, x_7)$,
fila i-ésima: $A_{i\cdot} = A(i,:)$,
columna j-ésima: $A_{\cdot j} = A(:,j)$,
parte de la fila i-ésima: $A(i, 1:4) = [\, a_{i1} \;\; a_{i2} \;\; a_{i3} \;\; a_{i4} \,]$
parte de la columna j-ésima: $A(2:4, j) = [\, a_{2j} \;\; a_{3j} \;\; a_{4j} \,]^{\mathrm{T}}$
submatriz: $A(3:6, 2:5)$.

2.3 Métodos ingenuos

Teóricamente, resolver el sistema $Ax = b$ es equivalente a la expresión

$$x = A^{-1}b.$$

Es claro que calcular la inversa de una matriz es mucho más dispendioso que resolver un sistema de ecuaciones; entonces, este camino sólo se utiliza en deducciones teóricas o, en muy raros casos, cuando A^{-1} se calcula muy fácilmente.

Otro método que podría utilizarse para resolver $Ax = b$ es la regla de Cramer. Para un sistema de orden 3 las fórmulas son:

$$x_1 = \frac{\det \begin{bmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{bmatrix}}{\det(A)}, \quad x_2 = \frac{\det \begin{bmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \end{bmatrix}}{\det(A)},$$

$$x_3 = \frac{\det \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{bmatrix}}{\det(A)}.$$

Supongamos ahora que cada determinante se calcula por medio de cofactores. Este cálculo se puede hacer utilizando cualquier fila o cualquier

columna; por ejemplo, si A es 3×3, utilizando la primera fila,

$$\det(A) = a_{11} \det \begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix} - a_{12} \det \begin{bmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{bmatrix} + a_{13} \det \begin{bmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}.$$

En general, sea M_{ij} la matriz $(n-1) \times (n-1)$ obtenida al suprimir de A la fila i y la columna j. Si se calcula $\det(A)$ utilizando la primera fila,

$$\det(A) = a_{11} \det(M_{11}) - a_{12} \det(M_{12}) + \cdots + (-1)^{(1+n)} a_{1n} \det(M_{1n}).$$

Sea μ_n el número de multiplicaciones necesarias para calcular, por cofactores, el determinante de una matriz de orden n. La fórmula anterior nos indica que

$$\mu_n > n\mu_{n-1}.$$

Como a su vez $\mu_{n-1} > (n-1)\mu_{n-2}$ y $\mu_{n-2} > (n-2)\mu_{n-3}$, ..., entonces

$$\begin{aligned} \mu_n &> n(n-1)(n-2)\cdots\mu_2 = n(n-1)(n-2)\cdots 2, \\ \mu_n &> n!\,. \end{aligned}$$

Para resolver un sistema de ecuaciones por la regla de Cramer, hay que calcular $n+1$ determinantes, luego el número total de multiplicaciones necesarias para resolver un sistema de ecuaciones por la regla de Cramer, calculando los determinantes por cofactores, es superior a $(n+1)!$.

Tomemos un sistema, relativamente pequeño, $n = 20$,

$$21! = 5.1091E19.$$

Siendo muy optimistas (sin tener en cuenta las sumas y otras operaciones concomitantes), supongamos que un computador del año 2000 hace 1000 millones de multiplicaciones por segundo. Entonces, el tiempo necesario para resolver un sistema de ecuaciones de orden 20 por la regla de Cramer y el método de cofactores es francamente inmanejable:

$$\text{tiempo } > 5.1091E10 \text{ segundos } = 16.2 \text{ siglos.}$$

2.4 Sistema diagonal

El caso más sencillo de (2.1) corresponde a una matriz diagonal. Para matrices triangulares, en particular para las diagonales, el determinante es el

producto de los n elementos diagonales. Entonces una matriz triangular es invertible si y solamente si todos los elementos diagonales son diferentes de cero.

La solución de un sistema diagonal se obtiene mediante

$$x_i = \frac{b_i}{a_{ii}}, \quad i = 1, ..., n. \tag{2.2}$$

Como los elementos diagonales son no nulos, no hay ningún problema para efectuar las divisiones.

2.5 Sistema triangular superior

Resolver un sistema triangular superior (A es triangular superior) es muy sencillo. Antes de ver el algoritmo en el caso general, veamos, por medio de un ejemplo, cómo se resuelve un sistema de orden 4.

Ejemplo 2.2. Resolver el siguiente sistema:

$$\begin{aligned} 4x_1 + 3x_2 - 2x_3 + x_4 &= 4 \\ -0.25x_2 + 2.5x_3 + 4.25x_4 &= -11 \\ 45x_3 + 79x_4 &= -203 \\ 2.8x_4 &= -5.6 \end{aligned}$$

De la cuarta ecuación, se deduce que $x_4 = -5.6/2.8 = -2$. A partir de la tercera ecuación

$$\begin{aligned} 45x_3 &= -203 - (79x_4) \\ x_3 &= \frac{-203 - (79x_4)}{45}. \end{aligned}$$

Reemplazando x_4 por su valor, se obtiene $x_3 = -1$. A partir de la segunda ecuación

$$\begin{aligned} -0.25x_2 &= -11 - (2.5x_3 + 4.25x_4) \\ x_2 &= \frac{-11 - (2.5x_3 + 4.25x_4)}{-0.25}. \end{aligned}$$

Reemplazando x_3 y x_4 por sus valores, se obtiene $x_2 = 0$. Finalmente, utilizando la primera ecuación,

$$\begin{aligned} 4x_1 &= 4 - (3x_2 - 2x_3 + x_4) \\ x_1 &= \frac{4 - (3x_2 - 2x_3 + x_4)}{4}. \end{aligned}$$

Reemplazando x_2, x_3 y x_4 por sus valores, se obtiene $x_1 = 1$. $\diamond$

En general, para resolver un sistema triangular, primero se calcula $x_n = b_n/a_{nn}$. Con este valor se puede calcular x_{n-1}, y así sucesivamente. Conocidos los valores x_{i+1}, x_{i+2}, ..., x_n, la ecuación i-ésima es

$$\begin{aligned} & a_{ii}x_i + a_{i,i+1}x_{i+1} + a_{i,i+2}x_{i+2} + ... + a_{in}x_n = b_i, \\ & a_{ii}x_i + A(i, i+1:n)\, x(i+1:n) = b_i, \\ & x_i = \frac{b_i - A(i, i+1:n)\, x(i+1:n)}{a_{ii}} \end{aligned}$$

Como se supone que A es regular (invertible o no singular), los elementos diagonales son no nulos y no se presentan problemas al efectuar la división.

El esquema del algoritmo es el siguiente:

$x_n = b_n/a_{nn}$
para $i = n-1, ..., 1$
    $x_i = (b_i - A(i, i+1:n)\, x(i+1:n))/a_{ii}$
fin-para

Esto se puede escribir en Scilab

```
x(n) = b(n)/a(n,n)
for i=n-1:-1:1
  x(i) = ( b(i) - a(i,i+1:n)*x(i+1:n) )/a(i,i)
end
```

La función completa podría ser así:

```
function [x, res] = solTriSup(a, b, eps)
  //
  // Solucion del sistema triangular superior  a x = b.
  //
  // a    es  una matriz triangular superior
  // b    es un vector columna
  // eps  es una valor positivo pequeno
  //      (parametro opcional).
```

```
    // res valdra   0  si el valor absoluto de un elemento
    //                 diagonal de  a  es menor o igual a  eps
    //              1  si todo funciono bien.
    // x  sera un vector columna con la solucion, si res = 1.
    //
    // Esta funcion trabaja unicamente con la parte triangular
    // superior de a y no verifica si realmente es triangular
    // superior.

    if argn(2) < 3, eps = 1.0e-10, end

    res = 0
    if min(abs(diag(a))) <= eps, return, end

    res = 1
    n = size(a,1)
    x = zeros(n,1)
    x(n) = b(n)/a(n,n)
    for k = n-1:-1:1
      x(k) = (b(k) - a(k,k+1:n)*x(k+1:n) )/a(k,k)
    end
  endfunction
```

Teniendo en cuenta las buenas características de Scilab, la función anterior se puede escribir un poco más corta. Sea `u = [ 2 3 5 7 11 13]'`. La orden `v = u(4:2)` produce un vector "vacío", es decir, `[ ]`. Además

```
s = 3.1 - u(4:2)*u(6:5)
```

asignará a `s` el valor 3.1. Entonces el cálculo de `x(n)` se puede hacer dentro del `for`:

```
    for k = n:-1:1
      x(k) = (b(k) - a(k,k+1:n)*x(k+1:n) )/a(k,k)
    end
```

2.5.1 Número de operaciones

Una de las maneras de medir la rapidez o lentitud de un método es mediante el conteo del número de operaciones. Usualmente se tienen en cuenta las sumas, restas, multiplicaciones y divisiones entre números de punto flotante,

aunque hay más operaciones fuera de las anteriores, por ejemplo las comparaciones y las operaciones entre enteros. Las cuatro operaciones se conocen con el nombre genérico de operaciones de punto flotante *flops* (floating point operations). Algunas veces se hacen dos grupos: por un lado sumas y restas, y por otro multiplicaciones y divisiones. Si se supone que el tiempo necesario para efectuar una multiplicación es bastante mayor que el tiempo de una suma, entonces se acostumbra a dar el número de multiplicaciones (o divisiones). El diseño de los procesadores actuales muestra tendencia al hecho de que los dos tiempos sean comparables. Entonces se acostumbra a evaluar el número de *flops*.

	Sumas y restas	Multiplicaciones y divisiones
cálculo de x_n	0	1
cálculo de x_{n-1}	1	2
cálculo de x_{n-2}	2	3
...		
cálculo de x_2	$n-2$	$n-1$
cálculo de x_1	$n-1$	n
Total	$n^2/2 - n/2$	$n^2/2 + n/2$

Número total de operaciones de punto flotante: n^2.

2.6 Sistema triangular inferior

La solución de un sistema triangular inferior $Ax = b$, A triangular inferior, es análoga al caso de un sistema triangular superior. Primero se calcula x_1, después x_2, enseguida x_3 y así sucesivamente hasta x_n.

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j}{a_{ii}}. \tag{2.3}$$

El esquema del algoritmo es el siguiente:

para $i = 1, ..., n$
    $x_i = (b_i - A(i, 1 : i-1)\, x(1 : i-1))/a_{ii}$
fin-para

El número de operaciones es exactamente el mismo del caso triangular superior.

2.7 Método de Gauss

El método de Gauss para resolver el sistema $Ax = b$ tiene dos partes; la primera es la triangularización del sistema, es decir, por medio de operaciones elementales, se construye un sistema

$$A'\, x = b', \tag{2.4}$$

equivalente al primero, tal que A' sea triangular superior. Que los sistemas sean equivalentes quiere decir que la solución de $Ax = b$ es exactamente la misma solución de $A'x = b'$. La segunda parte es simplemente la solución del sistema triangular superior.

Para una matriz, con índices entre 1 y n, el esquema de triangularización se puede escribir así:

para $k = 1, ..., n-1$
    buscar ceros en la columna k, por debajo de la diagonal.
fin-para k

Afinando un poco más:

para $k = 1, ..., n-1$
    para $i = k+1, ..., n$
        buscar ceros en la posición de a_{ik}.
    fin-para i
fin-para k

Ejemplo 2.3. Consideremos el siguiente sistema de ecuaciones:

$$\begin{aligned} 4x_1 + 3x_2 - 2x_3 + x_4 &= 4 \\ 3x_1 + 2x_2 + x_3 + 5x_4 &= -8 \\ -2x_1 + 3x_2 + x_3 + 2x_4 &= -7 \\ -5x_1 \qquad + x_3 + x_4 &= -8 \end{aligned}$$

En forma matricial se puede escribir:

$$\begin{bmatrix} 4 & 3 & -2 & 1 \\ 3 & 2 & 1 & 5 \\ -2 & 3 & 1 & 2 \\ -5 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 4 \\ -8 \\ -7 \\ -8 \end{bmatrix}$$

Es usual trabajar únicamente con los números, olvidando temporalmente los x_i. Más aún, se acostumbra trabajar con una matriz ampliada, resultado de pegar a la derecha de A el vector b.

$$\begin{bmatrix} 4 & 3 & -2 & 1 & 4 \\ 3 & 2 & 1 & 5 & -8 \\ -2 & 3 & 1 & 2 & -7 \\ -5 & 0 & 1 & 1 & -8 \end{bmatrix}$$

Inicialmente hay que buscar ceros en la primera columna. Para buscar cero en la posición `(2,1)`, fila 2 y columna 1, se hace la siguiente operación:

$$\text{fila2}_{\text{nueva}} \leftarrow \text{fila2}_{\text{vieja}} - (3/4)*\text{fila1}$$

Para hacer más sencilla la escritura la expresión anterior se escribirá simplemente:

$$\text{fila2} \leftarrow \text{fila2} - (3/4)*\text{fila1}$$

$$\begin{bmatrix} 4 & 3 & -2 & 1 & 4 \\ 0 & -0.25 & 2.5 & 4.25 & -11 \\ -2 & 3 & 1 & 2 & -7 \\ -5 & 0 & 1 & 1 & -8 \end{bmatrix}$$

Para buscar cero en la posición `(3,1)` se hace la siguiente operación:

$$\text{fila3} \leftarrow \text{fila3} - (-2/4)*\text{fila1}$$

$$\begin{bmatrix} 4 & 3 & -2 & 1 & 4 \\ 0 & -0.25 & 2.5 & 4.25 & -11 \\ 0 & 4.5 & 0 & 2.5 & -5 \\ -5 & 0 & 1 & 1 & -8 \end{bmatrix}$$

Para buscar cero en la posición `(4,1)` se hace la siguiente operación:

$$\text{fila4} \leftarrow \text{fila4} - (-5/4)*\text{fila1}$$

$$\begin{bmatrix} 4 & 3 & -2 & 1 & 4 \\ 0 & -0.25 & 2.5 & 4.25 & -11 \\ 0 & 4.5 & 0 & 2.5 & -5 \\ 0 & 3.75 & -1.5 & 2.25 & -3 \end{bmatrix}$$

Ahora hay que buscar ceros en la segunda columna. Para buscar cero en la posición `(3,2)` se hace la siguiente operación:

$$\text{fila3} \leftarrow \text{fila3} - (4.5/(-0.25))*\text{fila2}$$

$$\begin{bmatrix} 4 & 3 & -2 & 1 & 4 \\ 0 & -0.25 & 2.5 & 4.25 & -11 \\ 0 & 0 & 45 & 79 & -203 \\ 0 & 3.75 & -1.5 & 2.25 & -3 \end{bmatrix}$$

Para buscar cero en la posición `(4,2)` se hace siguiente operación:

$$\text{fila4} \leftarrow \text{fila4} - (3.75/(-0.25))*\text{fila2}$$

$$\begin{bmatrix} 4 & 3 & -2 & 1 & 4 \\ 0 & -0.25 & 2.5 & 4.25 & -11 \\ 0 & 0 & 45 & 79 & -203 \\ 0 & 0 & 36 & 66 & -168 \end{bmatrix}$$

Para buscar cero en la posición `(4,3)` se hace la siguiente operación:

$$\text{fila4} \leftarrow \text{fila4} - (36/45)*\text{fila3}$$

$$\begin{bmatrix} 4 & 3 & -2 & 1 & 4 \\ 0 & -0.25 & 2.5 & 4.25 & -11 \\ 0 & 0 & 45 & 79 & -203 \\ 0 & 0 & 0 & 2.8 & -5.6 \end{bmatrix}$$

El sistema resultante ya es triangular superior. Entonces se calcula primero $x_4 = -5.6/2.8 = -2$. Con este valor, utilizando la tercera ecuación resultante, se calcula x_3, después x_2 y x_1.

$$x = (1,\ 0,\ -1,\ -2).\ \diamond$$

De manera general, cuando ya hay ceros por debajo de la diagonal, en las columnas 1, 2, ..., $k-1$, para obtener cero en la posición (i,k) se hace la operación

$$\text{fila}i \leftarrow \text{fila}i - (a_{ik}/a_{kk})*\text{fila}k$$

Lo anterior se puede reescribir así:

$$\begin{aligned}
&\texttt{lik} = a_{ik}/a_{kk} \\
&A(i,:) = A(i,:) - \texttt{lik} * A(k,:) \\
&b_i = b_i - \texttt{lik} * b_k
\end{aligned}$$

Como en las columnas 1, 2, ..., $k-1$ hay ceros, tanto en la fila k como en la fila i, entonces a_{i1}, a_{i2}, ..., $a_{i,k-1}$ seguirán siendo cero. Además, las operaciones se hacen de tal manera que a_{ik} se vuelva cero. Entonces a_{ik} no se calcula puesto que dará 0. Luego los cálculos se hacen en la fila i a partir de la columna $k+1$.

$$\begin{aligned}
&\texttt{lik} = a_{ik}/a_{kk} \\
&a_{ik} = 0 \\
&A(i,k+1:n) = A(i,k+1:n) - \texttt{lik} * A(k,k+1:n) \\
&b_i = b_i - \texttt{lik} * b_k
\end{aligned}$$

En resumen, el esquema de la triangularización es:

para $k = 1, ..., n-1$
  para $i = k+1, ..., n$
    $\texttt{lik} = a_{ik}/a_{kk}, \quad a_{ik} = 0$
    $A(i,k+1:n) = A(i,k+1:n) - \texttt{lik} * A(k,k+1:n)$
    $b_i = b_i - \texttt{lik} * b_k$
  fin-para i
fin-para k

Este esquema funciona, siempre y cuando no aparezca un **pivote**, a_{kk}, nulo o casi nulo. Cuando aparezca es necesario buscar un elemento no nulo en el resto de la columna. Si, en el proceso de triangularización, toda la columna $A(k:n,k)$ es nula o casi nula, entonces A es singular.

para $k = 1, ..., n-1$
  para $i = k+1, ..., n$
    si $|a_{kk}| \leq \varepsilon$ **ent**
      buscar m, $k+1 \leq m \leq n$, tal que $|a_{mk}| > \varepsilon$
      si no fue posible **ent salir**
      intercambiar$(A(k, k:n), A(m, k:n))$
      intercambiar(b_k, b_m)
    fin-si
    $\texttt{lik} = a_{ik}/a_{kk}, \quad a_{ik} = 0$
    $A(i, k+1:n) = A(i, k+1:n) - \texttt{lik} * A(k, k+1:n)$
    $b_i = b_i - \texttt{lik} * b_k$
  fin-para i
fin-para k
si $|a_{nn}| \leq \varepsilon$ **ent salir**

Cuando en un proceso una variable toma valores enteros desde un límite inferior hasta un límite superior, y el límite inferior es mayor que el límite superior, el proceso no se efectúa.

Así, en el algoritmo anterior se puede hacer variar k, en el bucle externo, entre 1 y n, y entonces no es necesario controlar si $a_{nn} \approx 0$ ya que, cuando $k = n$, no es posible buscar m entre $n+1$ y n.

para $k = 1, ..., n$
  para $i = k+1, ..., n$
    si $|a_{kk}| \leq \varepsilon$ **ent**
      buscar m, $k+1 \leq m \leq n$, tal que $|a_{mk}| > \varepsilon$
      si no fue posible **ent salir**
      intercambiar$(A(k, k:n), A(m, k:n))$
      intercambiar(b_k, b_m)
    fin-si
    $\texttt{lik} = a_{ik}/a_{kk}, \quad a_{ik} = 0$
    $A(i, k+1:n) = A(i, k+1:n) - \texttt{lik} * A(k, k+1:n)$
    $b_i = b_i - \texttt{lik} * b_k$
  fin-para i
fin-para k

```
function [a, b, indic] = triangulariza(a, b, eps)
  // Triangulariza un sistema de ecuaciones
  // con matriz invertible.
```

```
  //
  // indic valdra  1  si todo funciono bien,
  //               0  si la matriz es singular o casi.
  //
  n = size(a,1)
  if argn(2) < 3, eps = 1.0e-10, end
  for k=1:n
    if abs(a(k,k)) <= eps
      m = posNoNulo(a, k)
      if m == 0
        indic = 0
        return
      end
      t = a(k,k:n)
      a(k,k:n) = a(m,k:n)
      a(m,k:n) = t
      t = b(k)
      b(k) = b(m)
      b(m) = t
    end
    for i=k+1:n
      lik = a(i,k)/a(k,k)
      a(i,k) = 0
      a(i,k+1:n) = a(i,k+1:n) - lik*a(k,k+1:n)
      b(i) = b(i) - lik*b(k)
    end
  end
  indic = 1
endfunction
//-------------------------------------------------------------
function m = posNoNulo(a, k, eps)
  // Busca la posicion del primer elemento no nulo en la
  // columna  k, debajo de la diagonal.
  //
  // Si no es posible encontrarlo,  m  valdra 0.
  //
  if argn(2) < 3, eps = 1.0e-10, end
  n = size(a,1)
  for i = k+1:n
    if abs(a(i,k)) >= eps
```

```
      m = i
      return
    end
  end
  m = 0
endfunction
//-----------------------------------------------------------
function [x, indic] = Gauss(a, b, eps)
  // Solucion de un sistema de ecuaciones
  // por el metododе Gauss.
  //
  // indic valdra  1  si todo funciono bien,
  //                  en este caso el vector columna x
  //                  sera la solucion.
  //               0  si la matriz es singular o casi
  //              -1  los tamanos son incompatibles.
  //
  indic = -1
  x = []
  n = verifTamanoAb(a, b)
  if n == 0, return, end

  if argn(2) < 3, eps = 1.0e-10, end

  indic = 0
  x = []
  [a, b, res] = triangulariza(a, b, eps)
  if res == 0, return, end

  indic = 1
  x = solTriSup(a, b, eps)
endfunction
//-----------------------------------------------------------
function n = verifTamanoAb(a, b)
  // Esta funcion verifica si los tamanos de  a, b
  // corresponden a un sistema cuadrado  a x = b.
  // Devuelve  n (num. de filas)  si todo esta bien,
  // devuelve  0  si hay errores.

  [n1, n2] = size(a)
```

```
  [n3, n4] = size(b)
  if n1 <> n2 | n1 <> n3 | n4 <> 1 | n1 < 1
    printf('\nTamanos inadecuados.\n\n')
    n = 0
  else
    n = n1
  end
endfunction
```

2.7.1 Número de operaciones

En el método de Gauss hay que tener en cuenta el número de operaciones de cada uno de los dos procesos: triangularización y solución del sistema triangular.

Triangularización

Consideremos inicialmente la búsqueda de cero en la posición $(2,1)$. Para efectuar $A(2,2:n) = A(2,2:n) - \texttt{lik} * A(1,2:n)$ es necesario hacer $n-1$ sumas y restas. Para $b_2 = b_2 - \texttt{lik} * b_1$ es necesario una resta. En resumen n sumas (o restas). Multiplicaciones y divisiones: una división para calcular `lik`; $n-1$ multiplicaciones para $\texttt{lik} * A(1,2:n)$ y una para $\texttt{lik} * b_1$. En resumen, $n+1$ multiplicaciones (o divisiones).

Para obtener un cero en la posición $(3,1)$ se necesita exactamente el mismo número de operaciones. Entonces para la obtener ceros en la primera columna:

	Sumas y restas	Multiplicaciones y divisiones
cero en la posición de a_{21}	n	$n+1$
cero en la posición de a_{31}	n	$n+1$
...		
cero en la posición de a_{n1}	n	$n+1$
Total para la columna 1	$(n-1)n$	$(n-1)(n+1)$

Un conteo semejante permite ver que se requieren $n-1$ sumas y n multiplicaciones para obtener un cero en la posición de a_{32}. Para buscar ceros en la columna 2 se van a necesitar $(n-2)(n-1)$ sumas y $(n-2)n$ multiplicaciones.

	Sumas y restas	Multiplicaciones y divisiones
ceros en la columna 1	$(n-1)n$	$(n-1)(n+1)$
ceros en la columna 2	$(n-2)(n-1)$	$(n-2)n$
ceros en la columna 3	$(n-3)(n-2)$	$(n-3)(n-1)$
...		
ceros en la columna $n-2$	2(3)	2(4)
ceros en la columna $n-1$	1(2)	1(3)

Es necesario utilizar el resultado

$$\sum_{i=1}^{m} i^2 = \frac{m(m+1)(2m+1)}{6}.$$

Número de sumas y restas:

$$\sum_{i=1}^{n-1} i(i+1) = \sum_{i=1}^{n-1}(i^2+i) = \frac{n^3}{3} - \frac{n}{3} \approx \frac{n^3}{3}.$$

Número de multiplicaciones y divisiones:

$$\sum_{i=1}^{n-1} i(i+2) = \sum_{i=1}^{n-1}(i^2+2i) = \frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6} \approx \frac{n^3}{3}.$$

Número de operaciones:

$$\frac{n^3}{3} - \frac{n}{3} + \frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6} = \frac{2n^3}{3} + \frac{n^2}{2} - \frac{7n}{6} \approx \frac{2n^3}{3}.$$

Proceso completo

El número de operaciones para las dos partes, triangularización y solución del sistema triangular, es

$$\frac{2n^3}{3} + \frac{3n^2}{2} - \frac{7n}{6} \approx \frac{2n^3}{3}.$$

Para valores grandes de n el número de operaciones de la solución del sistema triangular es despreciable con respecto al número de operaciones de la triangularización.

2.8 Factorización LU

Si durante el proceso del método de Gauss no fue necesario intercambiar filas, entonces se puede demostrar que se obtiene fácilmente la factorización $A = LU$, donde L es una matriz triangular inferior con unos en la diagonal y U es una matriz triangular superior. La matriz U es simplemente la matriz triangular superior obtenida al final del proceso.

Para el ejemplo anterior:

$$U = \begin{bmatrix} 4 & 3 & -2 & 1 \\ 0 & -0.25 & 2.5 & 4.25 \\ 0 & 0 & 45 & 79 \\ 0 & 0 & 0 & 2.8 \end{bmatrix}$$

La matriz L, con unos en la diagonal, va a estar formada simplemente por los coeficientes `lik`$= l_{ik} = a_{ik}/a_{kk}$.

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{bmatrix}$$

Siguiendo con el ejemplo:

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.75 & 1 & 0 & 0 \\ -0.5 & -18 & 1 & 0 \\ -1.25 & -15 & 0.8 & 1 \end{bmatrix}$$

En este ejemplo, fácilmente se comprueba que $LU = A$. Esta factorización es útil para resolver otro sistema $Ax = \tilde{b}$, exactamente con la misma matriz de coeficientes, pero con diferentes términos independientes.

$$\begin{aligned} Ax &= \tilde{b}, \\ LUx &= \tilde{b}, \\ Ly &= \tilde{b}, \\ \text{donde} \quad Ux &= y. \end{aligned}$$

En resumen:

- Resolver $Ly = \tilde{b}$ para obtener y.
- Resolver $Ux = y$ para obtener x.

Ejemplo 2.4. Resolver

$$\begin{aligned} 4x_1 + 3x_2 - 2x_3 + x_4 &= 8 \\ 3x_1 + 2x_2 + x_3 + 5x_4 &= 30 \\ -2x_1 + 3x_2 + x_3 + 2x_4 &= 15 \\ -5x_1 \qquad\quad + x_3 + x_4 &= 2 \end{aligned}$$

Al resolver

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.75 & 1 & 0 & 0 \\ -0.5 & -18 & 1 & 0 \\ -1.25 & -15 & 0.8 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 8 \\ 30 \\ 15 \\ 2 \end{bmatrix}$$

se obtiene $y = \begin{bmatrix} 8 & 24 & 451 & 11.2 \end{bmatrix}^{\mathrm{T}}$. Al resolver

$$\begin{bmatrix} 4 & 3 & -2 & 1 \\ 0 & -0.25 & 2.5 & 4.25 \\ 0 & 0 & 45 & 79 \\ 0 & 0 & 0 & 2.8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 8.0 \\ 24.0 \\ 451.0 \\ 11.2 \end{bmatrix}$$

se obtiene la solución final $x = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}^{\mathrm{T}}$. $\diamond$

Resolver un sistema triangular, con unos en la diagonal, requiere $n^2 - n \approx n^2$ operaciones. Entonces, para resolver un sistema adicional, con la misma matriz A, se requiere efectuar aproximadamente $2n^2$ operaciones, en lugar de $2n^3/3$ que se requerirían si se volviera a empezar el proceso.

La factorización $A = LU$ es un subproducto gratuito del método de Gauss; gratuito en tiempo y en requerimientos de memoria. No se requiere tiempo adicional puesto que el cálculo de los `lik` se hace dentro del método de Gauss. Tampoco se requiere memoria adicional puesto que los valores l_{ik} se pueden ir almacenando en A en el sitio de a_{ik} que justamente vale cero.

En el algoritmo hay únicamente un pequeño cambio:

$\vdots$
$\texttt{lik} = a_{ik}/a_{kk}$
$a_{ik} = \texttt{lik}$
$A(i, k+1 : n-1) = A(i, k+1 : n-1) - \texttt{lik} * A(k, k+1 : n-1)$
$b_i = b_i - \texttt{lik} * b_k$
$\vdots$

En la matriz final A estará la información indispensable de L y de U.

$$L = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ l_{21} & u_{22} & u_{23} & \cdots & u_{2n} \\ l_{31} & l_{32} & u_{31} & \cdots & u_{3n} \\ \vdots & & & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & u_{nn} \end{bmatrix}$$

En el ejemplo anterior, la matriz final con información de L y de U es:

$$\begin{bmatrix} 4 & 3 & -2 & 1 \\ 0.75 & -0.25 & 2.5 & 4.25 \\ -0.5 & -18 & 45 & 79 \\ -1.25 & -15 & 0.8 & 2.8 \end{bmatrix}$$

2.9 Método de Gauss con pivoteo parcial

En el método de Gauss clásico, únicamente se intercambian filas cuando el pivote, a_{kk}, es nulo o casi nulo. Como el pivote (el elemento a_{kk} en la iteración k) va a ser divisor para el cálculo de `lik`, y como el error de redondeo o de truncamiento se hace mayor cuando el divisor es cercano a cero, entonces es muy conveniente buscar que el pivote sea grande en valor absoluto. Es decir, hay que evitar los pivotes que sin ser nulos son cercanos a cero.

En el método de Gauss con pivoteo parcial se busca el elemento dominante, o sea, el de mayor valor absoluto en la columna k de la diagonal hacia abajo, es decir, entre los valores $|a_{kk}|$, $|a_{k+1,k}|$, $|a_{k+2,k}|$, ..., $|a_{kn}|$, y se intercambian la fila k y la fila del valor dominante. Esto mejora notablemente, en muchos casos, la precisión de la solución final obtenida.

Se dice que el pivoteo es total si en la iteración k se busca el mayor valor de $\{|a_{ij}| : k \leq i, j \leq n\}$. En este caso es necesario intercambiar dos filas

y dos columnas. Así se consigue mejorar un poco la precisión con respecto al método de pivoteo parcial, pero a un costo nada despreciable. En el método de pivoteo parcial se busca el mayor valor entre $n-k+1$ valores. En el pivoteo total se busca entre $(n-k+1)^2$ valores. Si se busca, de manera secuencial, el máximo entre p elementos, entonces hay que hacer, además de operaciones de asignación, por lo menos $p-1$ comparaciones. Estas operaciones no son de punto flotante y son más rápidas que ellas, pero para n grande, el tiempo utilizado no es despreciable. En el método de pivoteo parcial hay aproximadamente $n^2/2$ comparaciones, en el pivoteo total aproximadamente $n^3/6$. En resumen, con el pivoteo total se gana un poco de precisión, pero se gasta bastante más tiempo. El balance aconseja preferir el pivoteo parcial.

Ejemplo 2.5. Resolver por el método de Gauss con pivoteo parcial el siguiente sistema de ecuaciones.

$$\begin{aligned} 4x_1 + 3x_2 - 2x_3 + x_4 &= 4 \\ 3x_1 + 2x_2 + x_3 + 5x_4 &= -8 \\ -2x_1 + 3x_2 + x_3 + 2x_4 &= -7 \\ -5x_1 \qquad + x_3 + x_4 &= -8 \end{aligned}$$

La matriz aumentada es:

$$\begin{bmatrix} 4 & 3 & -2 & 1 & 4 \\ 3 & 2 & 1 & 5 & -8 \\ -2 & 3 & 1 & 2 & -7 \\ -5 & 0 & 1 & 1 & -8 \end{bmatrix}$$

El valor dominante de $A(1:4,1)$ es -5 y está en la fila 4. Entonces se intercambian las filas 1 y 4.

$$\begin{bmatrix} -5 & 0 & 1 & 1 & -8 \\ 3 & 2 & 1 & 5 & -8 \\ -2 & 3 & 1 & 2 & -7 \\ 4 & 3 & -2 & 1 & 4 \end{bmatrix}$$

Buscar ceros en las posiciones de $a_{21},\ a_{31},\ a_{41}$ se hace de la manera habitual usando los valores de `lik`$= 3/(-5) = -0.6,\ 0.4$ y -0.8. Se obtiene

$$\begin{bmatrix} -5 & 0 & 1 & 1 & -8 \\ 0 & 2 & 1.6 & 5.6 & -12.8 \\ 0 & 3 & 0.6 & 1.6 & -3.8 \\ 0 & 3 & -1.2 & 1.8 & -2.4 \end{bmatrix}$$

El valor dominante de $A(2:4,2)$ es 3 y está en la fila 3 (o en la fila 4). Entonces se intercambian las filas 2 y 3.

$$\begin{bmatrix} -5 & 0 & 1 & 1 & -8 \\ 0 & 3 & 0.6 & 1.6 & -3.8 \\ 0 & 2 & 1.6 & 5.6 & -12.8 \\ 0 & 3 & -1.2 & 1.8 & -2.4 \end{bmatrix}$$

Buscar ceros en las posiciones de a_{32}, a_{42} se hace usando los valores de `lik`$= 2/3 = 0.6666$ y 1. Se obtiene

$$\begin{bmatrix} -5 & 0 & 1 & 1 & -8 \\ 0 & 3 & 0.6 & 1.6 & -3.8 \\ 0 & 0 & 1.2 & 4.5333 & -10.2667 \\ 0 & 0 & -1.8 & 0.2 & 1.4 \end{bmatrix}$$

Hay que intercambiar las filas 3 y 4.

$$\begin{bmatrix} -5 & 0 & 1 & 1 & -8 \\ 0 & 3 & 0.6 & 1.6 & -3.8 \\ 0 & 0 & -1.8 & 0.2 & 1.4 \\ 0 & 0 & 1.2 & 4.5333 & -10.2667 \end{bmatrix}$$

El valor de `lik` es $1.2/(-1.8) = -0.6667$. Se obtiene

$$\begin{bmatrix} -5 & 0 & 1 & 1 & -8 \\ 0 & 3 & 0.6 & 1.6 & -3.8 \\ 0 & 0 & -1.8 & 0.2 & 1.4 \\ 0 & 0 & 0 & 4.6667 & -9.3333 \end{bmatrix}$$

Al resolver el sistema triangular superior, se encuentra la solución:

$$x = (1,\ 0,\ -1,\ -2)\,.\ \diamond$$

En Scilab la búsqueda del valor dominante y su fila se puede hacer mediante:

```
[vmax, posMax] = max(abs(a(k:n,k)))
m = k - 1 + posMax
if vmax <= eps, indic = 0, return, end
```

El ejemplo anterior sirve simplemente para mostrar el desarrollo del método de Gauss con pivoteo parcial, pero no muestra sus ventajas. El ejemplo

siguiente, tomado de [Atk78], se resuelve inicialmente por el método de Gauss sin pivoteo y después con pivoteo parcial. Los cálculos se hacen con cuatro cifras decimales.

$$\begin{aligned} 0.729x_1 + 0.81x_2 + 0.9x_3 &= 0.6867 \\ x_1 + x_2 + x_3 &= .8338 \\ 1.331x_1 + 1.21x_2 + 1.1x_3 &= 1 \end{aligned}$$

Con la solución exacta, tomada con cuatro cifras decimales, es

$$x = (\,0.2245,\ 0.2814,\ 0.3279\,).$$

Al resolver el sistema por el método de Gauss, con cuatro cifras decimales y sin pivoteo, resultan los siguientes pasos:

$$\begin{bmatrix} 0.7290 & 0.8100 & 0.9000 & 0.6867 \\ 1.0000 & 1.0000 & 1.0000 & 0.8338 \\ 1.3310 & 1.2100 & 1.1000 & 1.0000 \end{bmatrix}$$

Con `lik = 1.3717` y con `lik = 1.8258` se obtiene

$$\begin{bmatrix} 0.7290 & 0.8100 & 0.9000 & 0.6867 \\ 0.0000 & -0.1111 & -0.2345 & -0.1081 \\ 0.0000 & -0.2689 & -0.5432 & -0.2538 \end{bmatrix}$$

Con `lik = 2.4203` se obtiene

$$\begin{bmatrix} 0.7290 & 0.8100 & 0.9000 & 0.6867 \\ 0.0000 & -0.1111 & -0.2345 & -0.1081 \\ 0.0000 & 0.0000 & 0.0244 & 0.0078 \end{bmatrix}$$

La solución del sistema triangular da:

$$x = (\,0.2163,\ 0.2979,\ 0.3197\,).$$

Sea x^* la solución exacta del sistema $Ax = b$. Para comparar x^1 y x^2, dos aproximaciones de la solución, se miran sus distancias a x^*:

$$\|x^1 - x^*\|\ , \quad \|x^2 - x^*\|.$$

Si $\|x^1 - x^*\| < \|x^2 - x^*\|$, entonces x^1 es, entre x^1 y x^2, la mejor aproximación de x^*. Cuando no se conoce x^*, entonces se utiliza la norma del vector

residuo o resto, $r = Ax - b$. Si x es la solución exacta, entonces la norma de su resto vale cero. Entonces hay que comparar

$$\|Ax^1 - b\| \ , \quad \|Ax^2 - b\|.$$

Para la solución obtenida por el método de Gauss, sin pivoteo,

$$\|Ax - b\| = \texttt{1.0357e-004} \ , \quad \|x - x^*\| = 0.0202 \ .$$

En seguida está el método de Gauss con pivoteo parcial, haciendo cálculos con 4 cifras decimales.

$$\begin{bmatrix} 0.7290 & 0.8100 & 0.9000 & 0.6867 \\ 1.0000 & 1.0000 & 1.0000 & 0.8338 \\ 1.3310 & 1.2100 & 1.1000 & 1.0000 \end{bmatrix}$$

Intercambio de las filas 1 y 3.

$$\begin{bmatrix} 1.3310 & 1.2100 & 1.1000 & 1.0000 \\ 1.0000 & 1.0000 & 1.0000 & 0.8338 \\ 0.7290 & 0.8100 & 0.9000 & 0.6867 \end{bmatrix}$$

Con `lik = 0.7513` y con `lik = 0.5477` se obtiene

$$\begin{bmatrix} 1.3310 & 1.2100 & 1.1000 & 1.0000 \\ 0.0000 & 0.0909 & 0.1736 & 0.0825 \\ 0.0000 & 0.1473 & 0.2975 & 0.1390 \end{bmatrix}$$

Intercambio de las filas 2 y 3.

$$\begin{bmatrix} 1.3310 & 1.2100 & 1.1000 & 1.0000 \\ 0.0000 & 0.1473 & 0.2975 & 0.1390 \\ 0.0000 & 0.0909 & 0.1736 & 0.0825 \end{bmatrix}$$

Con `lik = 0.6171` se obtiene

$$\begin{bmatrix} 1.3310 & 1.2100 & 1.1000 & 1.0000 \\ 0.0000 & 0.1473 & 0.2975 & 0.1390 \\ 0.0000 & 0.0000 & -0.0100 & -0.0033 \end{bmatrix}$$

La solución del sistema triangular da:

$$x = (\, 0.2267, \ 0.2770, \ 0.3300 \,).$$

El cálculo del residuo y la comparación con la solución exacta da:

$$\|Ax - b\| = \texttt{1.5112e-004}\ , \quad \|x - x^*\| = 0.0053\ .$$

Se observa que para este ejemplo la norma del residuo es del mismo orden de magnitud que la norma del residuo correspondiente a la solución obtenida sin pivoteo, aunque algo mayor. La comparación directa con la solución exacta favorece notablemente al método de pivoteo parcial: 0.0053 y 0.0202, relación de 1 a 4 aproximadamente. Además, "visualmente" se observa la mejor calidad de la solución obtenida con pivoteo.

2.10 Factorización LU=PA

Si se aplica el método de Gauss con pivoteo parcial muy probablemente se hace por lo menos un intercambio de filas y no se puede obtener la factorización $A = LU$, pero sí se puede obtener la factorización

$$LU = PA.$$

Las matrices L y U tienen el mismo significado de la factorización LU. P es una matriz de permutación, es decir, se obtiene mediante permutación de filas de la matriz identidad I.

Si P y Q son matrices de permutación, entonces:

- PQ es una matriz de permutación.
- $P^{-1} = P^{\mathrm{T}}$ (P es ortogonal).
- PA es una permutación de las filas de A.
- AP es una permutación de las columnas de A.

Una matriz de permutación P se puede representar de manera más compacta por medio de un vector $p \in \mathbb{R}^n$ con la siguiente convención:

$$P_{i\cdot} = I_{p_i\cdot}.$$

En palabras, la fila i de P es simplemente la fila p_i de I. Obviamente p debe cumplir:

$$p_i \in \{1, 2, 3, ..., n\}\ \forall i$$
$$p_i \neq p_j\ \forall i \neq j.$$

Por ejemplo, $p = (2, 4, 3, 1)$ representa la matriz

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

De la misma forma que en la factorización LU, los valores l_{ik} se almacenan en el sitio donde se anula el valor a_{ik}. El vector p inicialmente es $(1, 2, 3, ..., n)$. A medida que se intercambian las filas de la matriz, se intercambian las componentes de p.

Ejemplo 2.6. Obtener la factorización $LU = PA$, donde

$$A = \begin{bmatrix} 4 & 3 & -2 & 1 \\ 3 & 2 & 1 & 5 \\ -2 & 3 & 1 & 2 \\ -5 & 0 & 1 & 1 \end{bmatrix}.$$

Inicialmente $p = (1, 2, 3, 4)$. Para buscar el mejor pivote, se intercambian las filas 1 y 4.

$$p = (4, 2, 3, 1), \quad \begin{bmatrix} -5 & 0 & 1 & 1 \\ 3 & 2 & 1 & 5 \\ -2 & 3 & 1 & 2 \\ 4 & 3 & -2 & 1 \end{bmatrix}.$$

Buscando ceros en la primera columna y almacenando allí los valores l_{ik} se obtiene:

$$\begin{bmatrix} -5 & 0 & 1 & 1 \\ -0.6 & 2 & 1.6 & 5.6 \\ 0.4 & 3 & 0.6 & 1.6 \\ -0.8 & 3 & -1.2 & 1.8 \end{bmatrix}.$$

Para buscar el mejor pivote, se intercambian las filas 2 y 3.

$$p = (4, 3, 2, 1), \quad \begin{bmatrix} -5 & 0 & 1 & 1 \\ 0.4 & 3 & 0.6 & 1.6 \\ -0.6 & 2 & 1.6 & 5.6 \\ -0.8 & 3 & -1.2 & 1.8 \end{bmatrix}.$$

Buscando ceros en la segunda columna y almacenando allí los valores l_{ik} se obtiene:

$$\begin{bmatrix} -5 & 0 & 1 & 1 \\ 0.4 & 3 & 0.6 & 1.6 \\ -0.6 & 0.6667 & 1.2 & 4.5333 \\ -0.8 & 1 & -1.8 & 0.2 \end{bmatrix}.$$

Para buscar el mejor pivote, se intercambian las filas 3 y 4.

$$p = (4, 3, 1, 2), \quad \begin{bmatrix} -5 & 0 & 1 & 1 \\ 0.4 & 3 & 0.6 & 1.6 \\ -0.8 & 1 & -1.8 & 0.2 \\ -0.6 & 0.6667 & 1.2 & 4.5333 \end{bmatrix}.$$

Buscando ceros en la tercera columna y almacenando allí los valores l_{ik} se obtiene:

$$\begin{bmatrix} -5 & 0 & 1 & 1 \\ 0.4 & 3 & 0.6 & 1.6 \\ -0.8 & 1 & -1.8 & 0.2 \\ -0.6 & 0.6667 & -0.6667 & 4.6667 \end{bmatrix}.$$

En esta última matriz y en el arreglo p está toda la información necesaria para obtener L, U, P. Entonces:

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.4 & 1 & 0 & 0 \\ -0.8 & 1 & 1 & 0 \\ -0.6 & 0.6667 & -0.6667 & 1 \end{bmatrix}.$$

$$U = \begin{bmatrix} -5 & 0 & 1 & 1 \\ 0 & 3 & 0.6 & 1.6 \\ 0 & 0 & -1.8 & 0.2 \\ 0 & 0 & 0 & 4.6667 \end{bmatrix}.$$

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \ \diamond$$

Si se desea resolver el sistema $Ax = b$ a partir de la descomposición $PA = LU$, se considera el sistema $P^{-1}LUx = b$, o sea, $P^{\mathrm{T}}LUx = b$. Sean $z = LUx$ y $y = Ux$. La solución de $Ax = b$ tiene tres pasos:

- Resolver $P^{\mathrm{T}}z = b$, o sea, $z = Pb$.
- Resolver $Ly = z$.
- Resolver $Ux = y$.

Ejemplo 2.7. Para la matriz A del ejemplo anterior, resolver $Ax = b$ con $b = [4 \ \ -8 \ \ -7 \ \ -8]^{\mathrm{T}}$.

$$z = Pb = \begin{bmatrix} -8 \\ -7 \\ 4 \\ -8 \end{bmatrix}$$

$$Ly = z \ \text{, entonces} \quad y = \begin{bmatrix} -8 \\ -3.8 \\ 1.4 \\ -9.3333 \end{bmatrix}$$

$$Ux = y \ \text{, entonces} \quad x = \begin{bmatrix} 1 \\ 0 \\ -1 \\ -2 \end{bmatrix} \quad \diamond$$

En Scilab, la factorización se puede obtener mediante la orden

```
[L, U, P] = lu(A)
```

2.11 Método de Cholesky

Este método sirve para resolver el sistema $Ax = b$ cuando la matriz A es *definida positiva* (también llamada positivamente definida). Este tipo de matrices se presenta en problemas específicos de ingeniería y física, principalmente.

2.11.1 Matrices definidas positivas

Una matriz simétrica es definida positiva si

$$x^{\mathrm{T}} A x > 0, \quad \forall\, x \in \mathbb{R}^n,\, x \neq 0. \tag{2.5}$$

Para una matriz cuadrada cualquiera,

$$\begin{aligned} x^{\mathrm{T}}Ax &= \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \\ &= \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{bmatrix} \\ &= \sum_{i=1}^{n} \sum_{j=i}^{n} a_{ij}x_i x_j. \end{aligned}$$

Si A es simétrica,

$$x^{\mathrm{T}}Ax = \sum_{i=1}^{n} a_{ii}x_i^2 + 2\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} a_{ij}x_i x_j.$$

Ejemplo 2.8. Sea I la matriz identidad de orden n. Entonces $x^{\mathrm{T}}Ix = x^{\mathrm{T}}x = \|x\|^2$. Luego la matriz I es definida positiva. $\diamond$

Ejemplo 2.9. Sea A la matriz nula de orden n. Entonces $x^{\mathrm{T}}\mathbf{0}\,x = 0$. Luego la matriz nula no es definida positiva. $\diamond$

Ejemplo 2.10. Sea

$$\begin{aligned} A &= \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}. \\ x^{\mathrm{T}}Ax &= x_1^2 + 5x_2^2 + 4x_1x_2 \\ &= x_1^2 + 4x_1x_2 + 4x_2^2 + x_2^2 \\ &= (x_1 + 2x_2)^2 + x_2^2. \end{aligned}$$

Obviamente $x^{\mathrm{T}}Ax \geq 0$. Además $x^{\mathrm{T}}Ax = 0$ si y solamente si los dos sumandos son nulos, es decir, si y solamente si $x_2 = 0$ y $x_1 = 0$, o sea, cuando $x = 0$. Luego A es definida positiva. $\diamond$

Ejemplo 2.11. Sea

$$\begin{aligned} A &= \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}. \\ x^{\mathrm{T}}Ax &= x_1^2 + 4x_2^2 + 4x_1x_2 \\ &= (x_1 + 2x_2)^2. \end{aligned}$$

Obviamente $x^{\mathrm{T}}Ax \geq 0$. Pero si $x = (6, -3)$, entonces $x^{\mathrm{T}}Ax = 0$. Luego A no es definida positiva. $\diamond$

Ejemplo 2.12. Sea

$$\begin{aligned} A &= \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}. \\ x^{\mathrm{T}}Ax &= x_1^2 + 3x_2^2 + 4x_1x_2 \\ &= (x_1 + 2x_2)^2 - x_2^2. \end{aligned}$$

Si $x = (6, -3)$, entonces $x^{\mathrm{T}}Ax = -9$. Luego A no es definida positiva. $\diamond$

Ejemplo 2.13. Sea

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}.$$

Como A no es simétrica, entonces no es definida positiva. $\diamond$

Sean $\lambda_1, \lambda_2, \ldots, \lambda_n$ los valores propios de A. Si A es simétrica, entonces todos sus valores propios son reales.

Sea δ_i el determinante de la submatriz de A, de tamaño $i \times i$, obtenida al quitar de A las filas $i+1$, $i+2$, ..., n y las columnas $i+1$, $i+2$, ..., n. O sea,

$$\begin{aligned} \delta_1 &= \det([a_{11}]) = a_{11}, \\ \delta_2 &= \det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \\ \delta_3 &= \det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{13} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \\ &\vdots \\ \delta_n &= \det(A). \end{aligned}$$

La definición 2.5 tiene relación directa con el nombre matriz definida positiva. Sin embargo, no es una manera fácil o práctica de saber cuándo una matriz simétrica es definida positiva, sobre todo si A es grande. El teorema siguiente presenta algunas de las caracterizaciones de las matrices definidas positivas. Para matrices pequeñas ($n \leq 4$) la caracterización por medio de los δ_i puede ser la de aplicación más sencilla. La última caracterización, llamada factorización de Cholesky, es la más adecuada para matrices grandes. En [Str86], [NoD88] y [Mor01] hay demostraciones y ejemplos.

Teorema 2.1. *Sea A simétrica. Las siguientes afirmaciones son equivalentes.*

- *A es definida positiva.*
- $\lambda_i > 0$, $\forall i$.
- $\delta_i > 0$, $\forall i$.
- *Existe U matriz triangular superior e invertible tal que* $A = U^T U$.

2.11.2 Factorización de Cholesky

Scilab tiene la función `chol` para obtener la factorizacón de Cholesky. Cuando no es posible aparecerá un mensaje de error.

```
a = [ 4 -6; -6 25]
u = chol(a)
```

Antes de estudiar el caso general, veamos la posible factorización para los ejemplos de la sección anterior.

La matriz identidad se puede escribir como $I = I^{\mathrm{T}} I$, siendo I triangular superior invertible. Luego existe la factorización de Cholesky para la matriz identidad.

Si existe la factorización de Cholesky de una matriz, al ser U y U^{T} invertibles, entonces A debe ser invertible. Luego la matriz nula no tiene factorización de Cholesky.

Sea

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}.$$

Entonces

$$\begin{bmatrix} u_{11} & 0 \\ u_{12} & u_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}$$

$$\begin{aligned} u_{11}^2 &= 1 \\ u_{11} u_{12} &= 2, \\ u_{12}^2 + u_{22}^2 &= 5 \end{aligned}$$

Se deduce que

$$\begin{aligned} u_{11} &= 1 \\ u_{12} &= 2, \\ u_{22} &= 1, \\ U &= \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}. \end{aligned}$$

Entonces existe la factorización de Cholesky de A.

Cuando se calculó u_{11} se hubiera podido tomar $u_{11} = -1$ y se hubiera podido obtener otra matriz U. Se puede demostrar que si se escogen los elementos diagonales u_{ii} positivos, entonces la factorización, cuando existe, es única.

Sea

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}.$$

Entonces

$$\begin{aligned} \begin{bmatrix} u_{11} & 0 \\ u_{12} & u_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix} &= \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \\ u_{11}^2 &= 1 \\ u_{11}u_{12} &= 2, \\ u_{12}^2 + u_{22}^2 &= 4 \end{aligned}$$

Se deduce que

$$\begin{aligned} u_{11} &= 1 \\ u_{12} &= 2, \\ u_{22} &= 0, \\ U &= \begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

Entonces, aunque existe U tal que $A = U^{\mathrm{T}}U$, sin embargo no existe la factorización de Cholesky de A ya que U no es invertible.

Sea

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}.$$

Entonces

$$\begin{bmatrix} u_{11} & 0 \\ u_{12} & u_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$$
$$\begin{aligned} u_{11}^2 &= 1 \\ u_{11}u_{12} &= 2, \\ u_{12}^2 + u_{22}^2 &= 3 \end{aligned}$$

Se deduce que

$$\begin{aligned} u_{11} &= 1 \\ u_{12} &= 2, \\ u_{22}^2 &= -1. \end{aligned}$$

Entonces no existe la factorización de Cholesky de A.

En el caso general,

$$\begin{bmatrix} u_{11} & & & & & & \\ \vdots & & & & & & \\ u_{1k} & \cdots & u_{kk} & & & & \\ \vdots & & & & & & \\ u_{1j} & \cdots & u_{kj} & \cdots & u_{jj} & & \\ \vdots & & & & & & \\ u_{1n} & \cdots & u_{kn} & \cdots & u_{jn} & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & \cdots & u_{1k} & \cdots & u_{1j} & \cdots & u_{1n} \\ & & & & & & \vdots \\ & & u_{kk} & \cdots & u_{kj} & \cdots & u_{kn} \\ & & & & & & \vdots \\ & & & & u_{jj} & \cdots & u_{jn} \\ & & & & & & \vdots \\ & & & & & & u_{nn} \end{bmatrix}$$

El producto de la fila 1 de U^{T} por la columna 1 de U da:

$$u_{11}^2 = a_{11}.$$

Luego

$$u_{11} = \sqrt{a_{11}}. \tag{2.6}$$

El producto de la fila 1 de U^{T} por la columna j de U da:

$$u_{11}u_{1j} = a_{1j}.$$

Luego

$$u_{1j} = \frac{a_{1j}}{u_{11}}, \qquad j = 2, ..., n. \tag{2.7}$$

Al hacer el producto de la fila 2 de U^{T} por la columna 2 de U, se puede calcular u_{22}. Al hacer el producto de la fila 2 de U^{T} por la columna j de

U, se puede calcular u_{2j}. Se observa que el cálculo de los elementos de U se hace fila por fila. Supongamos ahora que se conocen los elementos de las filas 1, 2, ..., $k-1$ de U y se desea calcular los elementos de la fila k de U. El producto de la fila k de U^{T} por la columna k de U da:

$$\begin{aligned} \sum_{i=1}^{k} u_{ik}^2 &= a_{kk} \\ \sum_{i=1}^{k-1} u_{ik}^2 + u_{kk}^2 &= a_{kk}. \end{aligned}$$

Luego

$$u_{kk} = \sqrt{a_{kk} - \sum_{i=1}^{k-1} u_{ik}^2}\ , \qquad k = 2, ..., n. \tag{2.8}$$

El producto de la fila k de U^{T} por la columna j de U da:

$$\sum_{i=1}^{k} u_{ik} u_{ij} = a_{kj}.$$

Luego

$$u_{kj} = \frac{a_{kj} - \sum_{i=1}^{k-1} u_{ik} u_{ij}}{u_{kk}}\ , \qquad k = 2, ..., n,\ \ j = k+1, ..., n. \tag{2.9}$$

Si consideramos que el valor de la sumatoria es 0 cuando el límite inferior es más grande que el límite superior, entonces las fórmulas 2.8 y 2.9 pueden ser usadas para $k = 1, ..., n$.

Ejemplo 2.14. Sea

$$A = \begin{bmatrix} 16 & -12 & 8 & -16 \\ -12 & 18 & -6 & 9 \\ 8 & -6 & 5 & -10 \\ -16 & 9 & -10 & 46 \end{bmatrix}.$$

$$u_{11} = \sqrt{16} = 4$$

$$
\begin{aligned}
u_{12} &= \frac{-12}{4} &&= -3 \\
u_{13} &= \frac{8}{4} &&= 2 \\
u_{14} &= \frac{-16}{4} &&= -4
\end{aligned}
$$

$$u_{22} = \sqrt{18 - (-3)^2} = 3$$

$$
\begin{aligned}
u_{23} &= \frac{-6 - (-3)(2)}{3} &&= 0 \\
u_{24} &= \frac{9 - (-3)(-4)}{3} &&= -1
\end{aligned}
$$

$$u_{33} = \sqrt{5 - (2^2 + 0^2)} = 1$$

$$u_{34} = \frac{-10 - (\ 2(-4) + 0(-1)\ )}{1} = -2$$

$$u_{44} = \sqrt{46 - (\ (-4)^2 + (-1)^2 + (-2)^2\ )} = 5\,.$$

Entonces,

$$U = \begin{bmatrix} 4 & -3 & 2 & -4 \\ 0 & 3 & 0 & -1 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 5 \end{bmatrix}. \ \diamond$$

La factorización de Cholesky no existe cuando en la fórmula 2.8 la cantidad dentro del radical es negativa o nula. Utilizando el producto entre matrices, las fórmulas 2.8 y 2.9 se pueden reescribir así:

$$
\begin{aligned}
t &= a_{kk} - U(1:k-1,k)^{\mathrm{T}}\, U(1:k-1,k), \\
u_{kk} &= \sqrt{t}, \\
u_{kj} &= \frac{a_{kj} - U(1:k-1,k)^{\mathrm{T}}\ U(1:k-1,j)}{u_{kk}}
\end{aligned}
$$

Para ahorrar espacio de memoria, los valores u_{kk} y u_{kj} se pueden almacenar sobre los antiguos valores de a_{kk} y a_{kj}. O sea, al empezar el algoritmo se tiene la matriz A. Al finalizar, en la parte triangular superior del espacio

ocupado por A estará U.

$$t = a_{kk} - U(1:k-1,k)^{\mathrm{T}}\, U(1:k-1,k), \tag{2.10}$$

$$a_{kk} = \sqrt{t}, \tag{2.11}$$

$$a_{kj} = \frac{a_{kj} - U(1:k-1,k)^{\mathrm{T}}\, U(1:k-1,j)}{a_{kk}} \tag{2.12}$$

El siguiente es el esquema del algoritmo para la factorización de Cholesky. Si acaba normalmente, la matriz A es definida positiva. Si en algún momento $t \leq \varepsilon$, entonces A no es definida positiva.

datos: A, ε
para $k = 1, ..., n$
    cálculo de t según (2.10)
    si $t \leq \varepsilon$ **ent salir**
    $a_{kk} = \sqrt{t}$
    para $j = k+1, ..., n$
        cálculo de a_{kj} según (2.12)
    fin-para j
fin-para k

La siguiente es la implementación en Scilab, utilizando las operaciones matriciales de Scilab:

```
function [U, ind] = Cholesky(A)
  //
  // Factorizacion de Cholesky.
  //
  // Trabaja unicamente con la parte triangular superior.
  //
  // ind = 1  si se obtuvo la factorizacion de Choleky
  //     = 0  si A no es definida positiva
  //
  //************
  eps = 1.0e-8
  //************

  n = size(A,1)
  U = zeros(n,n)
```

```
  for k = 1:n
    t = A(k,k) - U(1:k-1,k)'*U(1:k-1,k)
    if t <= eps
      printf('Matriz no definida positiva.\n')
      ind = 0
      return
    end
    U(k,k)= sqrt(t)
    for j = k+1:n
      U(k,j) = ( A(k,j) - U(1:k-1,k)'*U(1:k-1,j) )/U(k,k)
    end
  end
  ind = 1
endfunction
```

2.11.3 Número de operaciones de la factorización

Para el cálculo del número de operaciones supongamos que el tiempo necesario para calcular una raíz cuadrada es del mismo orden de magnitud que el tiempo de una multiplicación.

	Sumas y restas	Multiplicaciones, divisiones y raíces
cálculo de u_{11}	0	1
cálculo de u_{12}	0	1
cálculo de u_{1n}	0	1
cálculo de u_{22}	1	2
cálculo de u_{23}	1	2
cálculo de u_{2n}	1	2
...		
cálculo de u_{nn}	$n-1$	n

Agrupando por filas:

	Sumas y restas	Multiplicaciones, divisiones y raíces
cálculo de U_1.	$n(0)$	$n(1)$
cálculo de U_2.	$(n-1)1$	$(n-1)2$
cálculo de U_3.	$(n-2)2$	$(n-2)3$
...		
cálculo de U_n.	$1(n-1)$	$1(n)$

Número de sumas y restas:

$$\sum_{i=1}^{n-1}(n-i)i = \frac{n^3-n}{6} \approx \frac{n^3}{6}.$$

Número de multiplicaciones, divisiones y raíces:

$$\sum_{i=1}^{n}(n+1-i)i = \frac{n^3}{6} + \frac{n^2}{2} + \frac{n}{3} \approx \frac{n^3}{6}.$$

Número total de operaciones:

$$\frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6} \approx \frac{n^3}{3}.$$

2.11.4 Solución del sistema

Una vez obtenida la factorización de Cholesky, resolver $Ax = b$ es lo mismo que resolver $U^{\mathrm{T}}Ux = b$. Al hacer el cambio de variable $Ux = y$, la solución del sistema $Ax = b$ se convierte en

$$\text{resolver} \qquad U^{\mathrm{T}}y = b, \tag{2.13}$$

$$\text{resolver} \qquad Ux = y. \tag{2.14}$$

Resolver cada uno de los dos sistemas es muy fácil. El primero es triangular inferior, el segundo triangular superior. El número total de operaciones para resolver el sistema está dado por la factorización más la solución de dos sistemas triangulares.

$$\text{Número de operaciones } \approx \frac{n^3}{3} + 2\,n^2 \approx \frac{n^3}{3}.$$

Esto quiere decir que para valores grandes de n, resolver un sistema, con A definida positiva, por el método de Cholesky, gasta la mitad del tiempo requerido por el método de Gauss.

El método de Cholesky se utiliza para matrices definidas positivas. Pero no es necesario tratar de averiguar por otro criterio si la matriz es definida positiva. Simplemente se trata de obtener la factorización de Cholesky de A simétrica. Si fue posible, entonces A es definida positiva y se continúa con la solución de los dos sistemas triangulares. Si no fue posible obtener la factorización de Cholesky, entonces A no es definida positiva y no se puede aplicar el método de Cholesky para resolver $Ax = b$.

Ejemplo 2.15. Resolver

$$\begin{bmatrix} 16 & -12 & 8 \\ -12 & 18 & -6 \\ 8 & -6 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 76 \\ -66 \\ 46 \end{bmatrix}.$$

La factorización de Cholesky es posible (A es definida positiva):

$$U = \begin{bmatrix} 4 & -3 & 2 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

Al resolver $U^{\mathrm{T}}y = b$ se obtiene

$$y = (19,\ -3,\ 4).$$

Finalmente, al resolver $Ux = y$ se obtiene

$$x = (3,\ -1,\ 2).\ \diamond$$

La implementación en Scilab de la solución de un sistema con matriz simétrica y definida positiva se puede hacer por medio de una función que llama tres funciones:

```
function [x, info] = solCholesky(a, b)
  // Solucion de un sistema de ecuaciones por
  // el método de Cholesky
  //
  // Se supone que  a  es simetrica y se utiliza
  // unicamente la parte triangular superior de  a.
```

```
  //
  // info valdra  1  si  a  es definida positiva,
  //                 asi  x  sera un vector columna
  //                 con la solucion,
  //              0  si  a  no es definida positiva.
  //
  [a, info] = Cholesky(a)
  if info == 0, return, end
  y = sol_UT_y_b(a, b)
  x = solTriSup(a, y)
endfunction
```

La segunda función, `y = sol_UT_y_b(U, b)` resuelve el sistema $U^{\mathrm{T}}y = b$, pero se tiene la información de U. Si se sabe con certeza que la matriz es definida positiva, en lugar de `Cholesky`, es preferible usar la función de Scilab `chol` más eficiente.

2.12 Solución por mínimos cuadrados

Consideremos ahora un sistema de ecuaciones $Ax = b$, no necesariamente cuadrado, donde A es una matriz $m \times n$ cuyas columnas son linealmente independientes. Esto implica que hay más filas que columnas, $m \geq n$, y que además el rango de A es n. Es muy probable que este sistema no tenga solución, es decir, tal vez no existe x que cumpla exactamente las m igualdades. Se desea que

$$\begin{aligned} Ax &= b, \\ Ax - b &= 0, \\ \|Ax - b\| &= 0, \\ \|Ax - b\|_2 &= 0, \\ \|Ax - b\|_2^2 &= 0. \end{aligned}$$

Es posible que lo deseado no se cumpla, entonces se quiere que el incumplimiento (el error) sea lo más pequeño posible. Se desea minimizar esa cantidad,

$$\min \ \|Ax - b\|_2^2 \, . \tag{2.15}$$

El vector x que minimice $\|Ax-b\|_2^2$ se llama solución por mínimos cuadrados. Como se verá más adelante, tal x existe y es único (suponiendo que las columnas de A son linealmente independientes).

2.12.1 En Scilab

La orden para hallar por la solución por mínimos cuadrados es la misma que para resolver sistemas de ecuaciones cuadrados, a saber, `a\b`. Por ejemplo, para resolver el sistema

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \\ 6 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{“=”} \begin{bmatrix} 43 \\ 77 \\ 109 \end{bmatrix}$$

basta con

```
a = [ 2 3; 4 5; 7 6 ], b = [ 43 77 109 ]'
x = a\b
```

El resultado obtenido es

```
 x  =

    7.6019417
    9.3009709
```

2.12.2 Derivadas parciales

Con el ánimo de hacer más clara la deducción, supongamos que A es una matriz 4×3. Sea $f(x) = \|Ax - b\|_2^2$,

$$\begin{aligned} f(x) =& (a_{11}x_1 + a_{12}x_2 + a_{13}x_3 - b_1)^2 + (a_{21}x_1 + a_{22}x_2 + a_{23}x_3 - b_2)^2 + \\ & (a_{31}x_1 + a_{32}x_2 + a_{33}x_3 - b_3)^2 + (a_{41}x_1 + a_{42}x_2 + a_{43}x_3 - b_4)^2. \end{aligned}$$

Es posible que algunos de los lectores de este texto no conozcan el cálculo en varias variables. En este capítulo y en el siguiente se requiere saber calcular derivadas parciales. A continuación se presenta una breve introducción al cálculo (mecánico) de las derivadas parciales.

Sea φ una función de varias variables con valor real, $\varphi : \mathbb{R}^n \to \mathbb{R}$. Bajo ciertas condiciones de existencia, la derivada parcial de φ con respecto a x_i se obtiene considerando las otras variables como constantes y derivando $\varphi(x_1, x_2, ..., x_n)$ únicamente con respecto a x_i. Esta derivada parcial se denota

$$\frac{\partial \varphi}{\partial x_i}.$$

Evaluada en un punto específico $\bar{x}$, se denota

$$\frac{\partial \varphi}{\partial x_i}(\bar{x}) \cdot$$

Por ejemplo, si $\varphi(x_1, x_2, x_3, x_4) = (4x_1^3 + 6x_4)^9 + 5x_1x_2 + 8x_4$,

$$\begin{aligned}
\frac{\partial \varphi}{\partial x_1} &= 9(4x_1^3 + 6x_4)^8(12x_1^2) + 5x_2, \\
\frac{\partial \varphi}{\partial x_2} &= 5x_1, \\
\frac{\partial \varphi}{\partial x_3} &= 0, \\
\frac{\partial \varphi}{\partial x_4} &= 54(4x_1^3 + 6x_4)^8 + 8.
\end{aligned}$$

2.12.3 Ecuaciones normales

Para obtener el mínimo de f se requiere que las tres derivadas parciales, $\partial f/\partial x_1$, $\partial f/\partial x_2$ y $\partial f/\partial x_3$, sean nulas.

$$\begin{aligned}
\frac{\partial f}{\partial x_1} =& 2(a_{11}x_1 + a_{12}x_2 + a_{13}x_3 - b_1)a_{11} \\
&+ 2(a_{21}x_1 + a_{22}x_2 + a_{23}x_3 - b_2)a_{21} \\
&+ 2(a_{31}x_1 + a_{32}x_2 + a_{33}x_3 - b_3)a_{31} \\
&+ 2(a_{41}x_1 + a_{42}x_2 + a_{43}x_3 - b_4)a_{41}.
\end{aligned}$$

Escribiendo de manera matricial,

$$\begin{aligned}
\frac{\partial f}{\partial x_1} =& 2(A_1.x - b_1)a_{11} + 2(A_2.x - b_2)a_{21} + 2(A_3.x - b_3)a_{31} \\
&+ 2(A_4.x - b_4)a_{41}.
\end{aligned}$$

Si B es una matriz y u un vector columna, entonces $(Bu)_i = B_i.u$.

$$\begin{aligned}
\frac{\partial f}{\partial x_1} &= 2\Big(((Ax)_1 - b_1)a_{11} + ((Ax)_2 - b_2)a_{21} + ((Ax)_3 - b_3)a_{31} \\
&\quad +((Ax)_4 - b_4 a_{41}\Big), \\
&= 2\sum_{i=1}^{4}(Ax-b)_i\, a_{i1}, \\
&= 2\sum_{i=1}^{4}(A._1)_i(Ax-b)_i, \\
&= 2\sum_{i=1}^{4}(A^{\mathrm{T}}{}_1.)_i(Ax-b)_i, \\
&= 2A^{\mathrm{T}}{}_1.(Ax-b), \\
&= 2\big(A^{\mathrm{T}}(Ax-b)\big)_1
\end{aligned}$$

De manera semejante

$$\begin{aligned}
\frac{\partial f}{\partial x_2} &= 2\big(A^{\mathrm{T}}(Ax-b)\big)_2\,, \\
\frac{\partial f}{\partial x_3} &= 2\big(A^{\mathrm{T}}(Ax-b)\big)_3
\end{aligned}$$

Igualando a cero las tres derivadas parciales y quitando el 2 se tiene

$$\begin{aligned}
\big(A^{\mathrm{T}}(Ax-b)\big)_1 &= 0, \\
\big(A^{\mathrm{T}}(Ax-b)\big)_2 &= 0, \\
\big(A^{\mathrm{T}}(Ax-b)\big)_3 &= 0
\end{aligned}$$

Es decir,

$$\begin{aligned}
A^{\mathrm{T}}(Ax-b) &= 0, \\
A^{\mathrm{T}}\,A\,x &= A^{\mathrm{T}}\,b\,.
\end{aligned} \tag{2.16}$$

Las ecuaciones (2.16) se llaman **ecuaciones normales** para la solución (o seudosolución) de un sistema de ecuaciones por mínimos cuadrados.

La matriz $A^{\mathrm{T}}A$ es simétrica de tamaño $n \times n$. En general, si A es una matriz $m \times n$ de rango r, entonces $A^{\mathrm{T}}A$ también es de rango r (ver [Str86]). Como se supuso que el rango de A es n, entonces $A^{\mathrm{T}}A$ es invertible. Más aún, $A^{\mathrm{T}}A$ es definida positiva.

Por ser $A^{\mathrm{T}}A$ invertible, hay una única solución de (2.16), o sea, hay un solo vector x que hace que las derivadas parciales sean nulas. En general, las derivadas parciales nulas son simplemente una condición necesaria para obtener el mínimo de una función (también lo es para máximos o para puntos de silla), pero en este caso, como $A^{\mathrm{T}}A$ es definida positiva, f es convexa, y entonces anular las derivadas parciales se convierte en condición necesaria y suficiente para el mínimo.

En resumen, si las columnas de A son linealmente independientes, entonces la solución por mínimos cuadrados existe y es única. Para obtener la solución por mínimos cuadrados se resuelven las ecuaciones normales.

Como $A^{\mathrm{T}}A$ es definida positiva, (2.16) se puede resolver por el método de Cholesky. Si $m \geq n$ y al hacer la factorización de Cholesky resulta que $A^{\mathrm{T}}A$ no es definida positiva, entonces las columnas de A son linealmente dependientes.

Si el sistema $Ax = b$ tiene solución exacta, ésta coincide con la solución por mínimos cuadrados.

Ejemplo 2.16. Resolver por mínimos cuadrados:

$$\begin{bmatrix} 2 & 1 & 0 \\ -1 & -2 & 3 \\ -2 & 2 & 1 \\ 5 & 4 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3.1 \\ 8.9 \\ -3.1 \\ 0.1 \end{bmatrix}.$$

Las ecuaciones normales dan:

$$\begin{bmatrix} 34 & 20 & -15 \\ 20 & 25 & -12 \\ -15 & -12 & 14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4.0 \\ -20.5 \\ 23.4 \end{bmatrix}$$

La solución por mínimos cuadrados es:

$$x = (2.0252,\ -1.0132,\ 2.9728)\,.$$

El error, $Ax - b$, es:

$$\begin{bmatrix} -0.0628 \\ 0.0196 \\ -0.0039 \\ 0.0275 \end{bmatrix}. \quad \diamond$$

Ejemplo 2.17. Resolver por mínimos cuadrados:

$$\begin{bmatrix} 2 & 1 & 3 \\ -1 & -2 & 0 \\ -2 & 2 & -6 \\ 5 & 4 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \\ -3 \\ 0 \end{bmatrix}.$$

Las ecuaciones normales dan:

$$\begin{bmatrix} 34 & 20 & 48 \\ 20 & 25 & 15 \\ 48 & 15 & 81 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -21 \\ 27 \end{bmatrix}$$

Al tratar de resolver este sistema de ecuaciones por el método de Cholesky; no se puede obtener la factorización de Cholesky, luego $A^{\mathrm{T}}A$ no es definida positiva, es decir, las columnas de A son linealmente dependientes. Si se aplica el método de Gauss, se obtiene que $A^{\mathrm{T}}A$ es singular y se concluye que las columnas de A son linealmente dependientes. $\diamond$

Ejemplo 2.18. Resolver por mínimos cuadrados:

$$\begin{bmatrix} 2 & 1 \\ -1 & -2 \\ -2 & 2 \\ 5 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ -6 \\ 6 \end{bmatrix}.$$

Las ecuaciones normales dan:

$$\begin{bmatrix} 34 & 20 \\ 20 & 25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 48 \\ 15 \end{bmatrix}$$

La solución por mínimos cuadrados es:

$$x = (2,\ -1)\,.$$

El error, $Ax - b$, es:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

En este caso, el sistema inicial tenía solución exacta y la solución por mínimos cuadrados coincide con ella. $\diamond$

La solución por mínimos cuadrados de un sistema sobredeterminado también se puede hacer en Scilab mediante `(a'*a)\(a'*b)` o por medio de `pinv(a)*b`, pero ambas son menos eficientes que `a\b`.

La implementación eficiente de la solución por mínimos cuadrados, vía ecuaciones normales, debe tener en cuenta algunos detalles. No es necesario construir toda la matriz simétrica $A^{\mathrm{T}}A$ (n^2 elementos). Basta con almacenar en un arreglo de tamaño $n(n+1)/2$ la parte triangular superior de $A^{\mathrm{T}}A$.

Este almacenamiento puede ser por filas, es decir, primero los n elementos de la primera fila, enseguida los $n-1$ elementos de la segunda fila a partir del elemento diagonal, después los $n-2$ de la tercera fila a partir del elemento diagonal y así sucesivamente hasta almacenar un solo elemento de la fila n. Si se almacena la parte triangular superior de $A^{\mathrm{T}}A$ por columnas, se almacena primero un elemento de la primera columna, enseguida dos elementos de la segunda columna y así sucesivamente. Cada una de las dos formas tiene sus ventajas y desventajas. La solución por el método de Cholesky debe tener en cuenta este tipo de estructura de almacenamiento de la información.

Otros métodos eficientes para resolver sistemas de ecuaciones por mínimos cuadrados utilizan matrices ortogonales de Givens o de Householder.

2.13 Sistemas tridiagonales

Un sistema $Ax = b$ se llama tridiagonal si la matriz A es tridiagonal, o sea, si

$$a_{ij} = 0 \quad \text{si} \quad |i-j| > 1,$$

es decir, A es de la forma

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & 0 & & 0 \\ 0 & a_{32} & a_{33} & a_{34} & & 0 \\ 0 & 0 & a_{43} & a_{44} & & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix}.$$

Estos sistemas se presentan en algunos problemas particulares, por ejemplo, al resolver, mediante diferencias finitas, una ecuación diferencial lineal de segundo orden con condiciones de frontera o en el cálculo de los coeficientes de un trazador cúbico ("spline").

Obviamente este sistema se puede resolver mediante el método de Gauss. Pero dadas las características especiales es mucho más eficiente sacar provecho de ellas. Se puede mostrar que si A admite descomposición LU, entonces estas dos matrices también guardan la estructura de A, es decir, L, además de ser triangular inferior, tiene ceros por debajo de la "subdiagonal" y U, además de ser triangular superior, tiene ceros por encima de la "superdiagonal".

Para simplificar, denotemos con f_i los elementos de la suddiagonal de L, d_i los elementos de la diagonal de U y u_i los elementos de la superdiagonal de U. Se conoce A y se desea conocer L y U a partir de la siguiente igualdad:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ f_1 & 1 & 0 & 0 & & 0 & 0 \\ 0 & f_2 & 1 & 0 & & 0 & 0 \\ 0 & 0 & f_3 & 1 & & 0 & 0 \\ & & & & \ddots & & \\ 0 & 0 & 0 & 0 & & 1 & 0 \\ 0 & 0 & 0 & 0 & & f_{n-1} & 1 \end{bmatrix} \begin{bmatrix} d_1 & u_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & d_2 & u_2 & 0 & & 0 & 0 \\ 0 & 0 & d_3 & u_3 & & 0 & 0 \\ 0 & 0 & 0 & d_4 & & 0 & 0 \\ & & & & \ddots & & \\ 0 & 0 & 0 & 0 & & d_{n-1} & u_{n-1} \\ 0 & 0 & 0 & 0 & & 0 & d_n \end{bmatrix} = A \ .$$

Sean F_i la fila i de L y C_j la columna j de U. Entonces los productos de las filas de L por las columnas de U producen las siguientes igualdades:

$$\begin{aligned} F_1C_1 : &\quad d_1 = a_{11} \\ F_1C_2 : &\quad u_1 = a_{12} \\ F_2C_1 : &\quad f_1d_1 = a_{21} \\ F_2C_2 : &\quad f_1u_1 + d_2 = a_{22} \\ F_2C_3 : &\quad u_2 = a_{23} \\ F_3C_2 : &\quad f_2d_2 = a_{32} \\ F_3C_3 : &\quad f_2u_2 + d_3 = a_{33} \\ F_3C_4 : &\quad u_3 = a_{34} \\ &\vdots \\ F_iC_{i-1} : &\quad f_{i-1}d_{i-1} = a_{i,i-1} \\ F_iC_i : &\quad f_{i-1}u_{i-1} + d_i = a_{ii} \\ F_iC_{i+1} : &\quad u_i = a_{i,i+1} \end{aligned}$$

A partir de las igualdades anteriores se obtienen los valores u_i, f_i y d_i:

$$\begin{aligned} d_1 &= a_{11}, \\ u_i &= a_{i,i+1}\,, \qquad i = 1, ..., n-1, \\ f_i &= \frac{a_{i+1,i}}{d_i}, \\ d_{i+1} &= a_{i+1,i+1} - f_i u_i \end{aligned} \tag{2.17}$$

Resolver $Ax = b$ es equivalente a resolver $LUx = b$. Entonces, si $Ux = y$, se resuelve $Ly = b$ y después $Ux = y$. Al explicitar las anteriores igualdades se tiene:

$$\begin{aligned} y_1 &= b_1, \\ f_{i-1}y_{i-1} + y_i &= b_i, \\ d_n x_n &= y_n, \\ d_i x_i + u_i x_{i+1} &= y_i\,. \end{aligned}$$

Las fórmulas explícitas son:

$$\begin{aligned} y_1 &= b_1, \\ y_i &= b_i - f_{i-1}y_{i-1}, \quad i = 2, ..., n, \\ x_n &= \frac{y_n}{d_n}, \\ x_i &= \frac{y_i - u_i x_{i+1}}{d_i}\,, \qquad i = n-1, n-2, ..., 1. \end{aligned} \tag{2.18}$$

Ejemplo 2.19. Resolver el sistema $Ax = b$, con

$$A = \begin{bmatrix} 2 & 4 & 0 & 0 \\ 3 & 5 & 6 & 0 \\ 0 & -4 & -5 & 1 \\ 0 & 0 & -1 & -2 \end{bmatrix}, \qquad b = \begin{bmatrix} -8 \\ 1 \\ -2 \\ -10 \end{bmatrix}.$$

Entonces

$$\begin{aligned}
d_1 &= 2\,,\\
u_1 &= 4\,,\\
f_1 &= \frac{3}{2} = 1.5\,,\\
d_2 &= 5 - 1.5 \times 4 = -1\,,\\
u_2 &= 6\,,\\
f_2 &= \frac{-4}{-1} = 4\,,\\
d_3 &= -5 - 4 \times 6 = -29\,,\\
u_3 &= 1,\\
f_3 &= \frac{-1}{-29} = 0.034483\,,\\
d_4 &= -2 - 0.034483 \times 1 = -2.034483\,,
\end{aligned}$$

Ahora la solución de los sistemas $Ly = b$, $Ux = y$:

$$\begin{aligned}
y_1 &= -8,\\
y_2 &= 1 - 1.5 \times (-8) = 13\,,\\
y_3 &= -2 - 4 \times 13 = -54\,,\\
y_4 &= -10 - 0.034483 \times -54 = -8.137931\,,\\
x_4 &= \frac{-8.137931}{-2.034483} = 4\,,\\
x_3 &= \frac{-54 - 1 \times 4}{-29} = 2\,,\\
x_2 &= \frac{13 - 6 \times 2}{-1} = -1\,,\\
x_1 &= \frac{-8 - 4 \times (-1)}{2} = -2\,. \ \diamond
\end{aligned}$$

Las fórmulas (2.17) y (2.18) se pueden utilizar sin ningún problema si todos los d_i son no nulos. Algún elemento diagonal de U resulta nulo si la matriz A no es invertible o si simplemente A no tiene factorización LU.

Ejemplo 2.20. Consideremos las dos matrices siguientes:

$$A = \begin{bmatrix} 2 & -3 \\ -8 & 12 \end{bmatrix}, \qquad A' = \begin{bmatrix} 0 & 2 \\ 3 & 4 \end{bmatrix}.$$

La matriz A no es invertible y d_2 resulta nulo. La matriz A' es invertible pero no tiene factorización LU. En este último caso, se obtiene $d_1 = 0$. $\diamond$

Si la matriz A es grande no se justifica almacenar todos los n^2 elementos. Basta con almacenar la diagonal, la subdiagonal y la superdiagonal, es decir $3n - 2$ números. Mejor aún, en el mismo sitio donde inicialmente se almacenan los elementos diagonales de A se pueden almacenar los elementos diagonales de U a medida que se van calculando, donde se almacenan los elementos subdiagonales de A se pueden almacenar los elementos subdiagoanles de L, los elementos superdiagonales de A son los mismos elementos superdiagonales de U, donde se almacena b se puede almacenar y y posteriormente x.

En resumen, una implementación eficiciente utiliza 4 vectores d, f, u y b. El primero y el cuarto están en $\mathbb{R}^n$, los otros dos están en $\mathbb{R}^{n-1}$. Al comienzo d, f, u contienen datos de A y los términos independientes están en b. Al final d, f, u contienen datos de L, U y la solución final (los x_i) estará en b.

SOLUCIÓN DE SISTEMA TRIDIAGONAL

datos: d, f, u, b, ε

si $|d_1| \leq \varepsilon$ **ent parar**
para $i = 1, ..., n-1$
  $f_i = \dfrac{f_i}{d_i}$
  $d_{i+1} = d_{i+1} - f_i * u_i$
  si $|d_{i+1}| \leq \varepsilon$ **ent parar**
fin-para
para $i = 2, ..., n$
  $b_i = b_i - f_{i-1} b_{i-1}$
fin-para
$b_n = \dfrac{b_n}{d_n}$
para $i = n-1, n-2, ..., 1$
  $b_i = \dfrac{b_i - u_i b_{i+1}}{d_i}$
fin-para

2.14 Cálculo de la inversa

En la mayoría de los casos **no es necesario calcular explícitamente la inversa** de una matriz, pues basta con resolver un sistema de ecuaciones. De todas formas, algunas pocas veces es indispensable obtener la inversa.

A continuación está el algoritmo para el cáculo de la inversa, tomado y adaptado de [Stewart 98], basado en la factorización $LU = PA$ (con pivoteo parcial). Se utiliza un vector p en $\mathbb{Z}^{n-1}$ que tiene toda la información indispensable para obtener la matriz P, pero no representa directamente la permutación. Al principio p es simplemente $(1, 2, ..., n-1)$.

Sólamente se utiliza memoria para una matriz. Al principio está A; al final del algoritmo, si `indic` $= 1$, está la inversa. Cuando `indic` $= 0$, la matriz es singular o casi singular.

Se utiliza la notación de Matlab y Scilab para las submatrices de A. Para los elementos de A y p se utiliza la notación usual con subíndices.

datos: A, ε
resultados: la inversa almacenada en A, `indic`

FACTORIZACIÓN:
$p = (1, 2, ..., n-1)$
para $k = 1 : n-1$
    determinar m tal que $|a_{mk}| = \max\{\ |a_{ik}| : i = k, ..., n\}$
    si $|a_{mk}| \leq \varepsilon$
        `indic = 0`, **parar**
    fin-si
    $p_k = m$
    si $m > k$
        $A(k, :) \leftrightarrow A(m, :)$
    fin-si
    $A(k+1 : n, k) = A(k+1 : n, k)/a_{kk}$
    $A(k+1 : n, k+1 : n) = A(k+1 : n, k+1 : n) - A(k+1 : n, k)A(k, k+1 : n)$
fin-para
si $|a_{nn}| \leq \varepsilon$
    `indic = 0`, **parar**
fin-si
`indic = 1`

CÁLCULO DE U^{-1}:
para $k = 1 : n$
  $a_{kk} = 1/a_{kk}$
  para $i = 1 : k-1$
    $a_{ik} = -a_{kk} A(i, i : k-1) A(i : k-1, k)$
  fin-para
fin-para

CÁLCULO DE $U^{-1}L^{-1}$:
para $k = n-1 : -1 : 1$
  $t = A(k+1 : n, k)$
  $A(k+1 : n, k) = 0$
  $A(:, k) = A(:, k) - A(:, k+1 : n)\ t$
fin-para

REORDENAMIENTO DE COLUMNAS:
para $k = n-1 : -1 : 1$
  si $p_k \neq k$
    $A(:, k) \leftrightarrow A(:, p_k)$
  fin-si
fin-para

Ejemplo 2.21.
```
A inicial
   -2.0000   -4.0000    4.0000   -2.0000
   -5.0000    1.0000    2.0000    1.0000
    4.0000   -3.0000    0.0000   -4.0000
   -2.0000   -3.0000    1.0000   -1.0000
p inicial :
   1   2   3

Factorisacion

k  = 1
m  = 2
p :
   2   2   3
intercambio de filas : 1  2
A despues de intercambio
   -5.0000    1.0000    2.0000    1.0000
   -2.0000   -4.0000    4.0000   -2.0000
```

```
    4.0000   -3.0000    0.0000   -4.0000
   -2.0000   -3.0000    1.0000   -1.0000
A despues de operaciones
   -5.0000    1.0000    2.0000    1.0000
    0.4000   -4.4000    3.2000   -2.4000
   -0.8000   -2.2000    1.6000   -3.2000
    0.4000   -3.4000    0.2000   -1.4000

k  = 2
m  = 2
p :
   2   2   3
A despues de operaciones
   -5.0000    1.0000    2.0000    1.0000
    0.4000   -4.4000    3.2000   -2.4000
   -0.8000    0.5000    0.0000   -2.0000
    0.4000    0.7727   -2.2727    0.4545

k  = 3
m  = 4
p :
   2   2   4
intercambio de filas : 3  4
A despues de intercambio
   -5.0000    1.0000    2.0000    1.0000
    0.4000   -4.4000    3.2000   -2.4000
    0.4000    0.7727   -2.2727    0.4545
   -0.8000    0.5000    0.0000   -2.0000
A despues de operaciones
   -5.0000    1.0000    2.0000    1.0000
    0.4000   -4.4000    3.2000   -2.4000
    0.4000    0.7727   -2.2727    0.4545
   -0.8000    0.5000   -0.0000   -2.0000

A despues de calcular inv. de U
   -0.2000   -0.0455   -0.2400   -0.1000
    0.4000   -0.2273   -0.3200    0.2000
    0.4000    0.7727   -0.4400   -0.1000
   -0.8000    0.5000   -0.0000   -0.5000
```

```
A despues de calcular U1*L1
   -0.2600    0.1900   -0.2400   -0.1000
    0.3200   -0.0800   -0.3200    0.2000
   -0.0600    0.3900   -0.4400   -0.1000
   -0.5000    0.2500    0.0000   -0.5000

inversa: despues de reordenamiento
    0.1900   -0.2600   -0.1000   -0.2400
   -0.0800    0.3200    0.2000   -0.3200
    0.3900   -0.0600   -0.1000   -0.4400
    0.2500   -0.5000   -0.5000    0.0000

Expresiones explicitas de L, U, P
L
    1.0000    0.0000    0.0000    0.0000
    0.4000    1.0000    0.0000    0.0000
    0.4000    0.7727    1.0000    0.0000
   -0.8000    0.5000   -0.0000    1.0000
U
   -5.0000    1.0000    2.0000    1.0000
    0.0000   -4.4000    3.2000   -2.4000
    0.0000    0.0000   -2.2727    0.4545
    0.0000    0.0000    0.0000   -2.0000
P :
   0   1   0   0
   1   0   0   0
   0   0   0   1
   0   0   1   0
```

3

Métodos iterativos

Los métodos de Gauss y Cholesky hacen parte de los métodos directos o finitos. Al cabo de un número finito de operaciones, en ausencia de errores de redondeo, se obtiene x^* solución del sistema $Ax = b$.

El método de Jacobi, Gauss-Seidel, SOR (sobrerrelajación), hacen parte de los métodos llamados indirectos o iterativos. En ellos se comienza con $x^0 = (x_1^0, x_2^0, ..., x_n^0)$, una aproximación inicial de la solución. A partir de x^0 se construye una nueva aproximación de la solución, $x^1 = (x_1^1, x_2^1, ..., x_n^1)$. A partir de x^1 se construye x^2 (aquí el superíndice indica la iteración y no indica una potencia). Así sucesivamente se construye una sucesión de vectores $\{x^k\}$, con el objetivo, no siempre garantizado, de que

$$\lim_{k \to \infty} x^k = x^*.$$

Generalmente los métodos indirectos son una buena opción cuando la matriz es muy grande y dispersa o rala (*sparse*), es decir, cuando el número de elementos no nulos es pequeño comparado con n^2, número total de elementos de A. En estos casos se debe utilizar una estructura de datos adecuada que permita almacenar únicamente los elementos no nulos.

3.1 Método de Gauss-Seidel

En cada iteración del método de Gauss-Seidel, hay n subiteraciones. En la primera subiteración se modifica únicamente x_1. Las demás coordenadas x_2, x_3, ..., x_n no se modifican. El cálculo de x_1 se hace de tal manera que

se satisfaga la primera ecuación.

$$\begin{aligned} x_1^1 &= \frac{b_1 - (a_{12}x_2^0 + a_{13}x_3^0 + \cdots + a_{1n}x_n^0)}{a_{11}}, \\ x_i^1 &= x_i^0\,, \quad i = 2, ..., n. \end{aligned}$$

En la segunda subiteración se modifica únicamente x_2. Las demás coordenadas x_1, x_3, ..., x_n no se modifican. El cálculo de x_2 se hace de tal manera que se satisfaga la segunda ecuación.

$$\begin{aligned} x_2^2 &= \frac{b_2 - (a_{21}x_1^1 + a_{23}x_3^1 + \cdots + a_{2n}x_n^1)}{a_{22}}, \\ x_i^2 &= x_i^1\,, \quad i = 1, 3, ..., n. \end{aligned}$$

Así sucesivamente, en la n-ésima subiteración se modifica únicamente x_n. Las demás coordenadas x_1, x_2, ..., x_{n-1} no se modifican. El cálculo de x_n se hace de tal manera que se satisfaga la n-ésima ecuación.

$$\begin{aligned} x_n^n &= \frac{b_n - (a_{n1}x_1^{n-1} + a_{n3}x_3^{n-1} + \cdots + a_{nn}x_n^{n-1})}{a_{nn}}, \\ x_i^n &= x_i^{n-1}\,, \quad i = 1, 2, ..., n-1. \end{aligned}$$

Ejemplo 3.1. Resolver

$$\begin{bmatrix} 10 & 2 & -1 & 0 \\ 1 & 20 & -2 & 3 \\ -2 & 1 & 30 & 0 \\ 1 & 2 & 3 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 26 \\ -15 \\ 53 \\ 47 \end{bmatrix}$$

partiendo de $x^0 = (1,\ 2,\ 3,\ 4)$.

$$x_1^1 = \frac{26 - (2 \times 2 + (-1) \times 3 + 0 \times 4)}{10} = 2.5,$$
$$x^1 = (2.5,\ 2,\ 3,\ 4).$$
$$x_2^2 = \frac{-15 - (1 \times 2.5 + (-2) \times 3 + 3 \times 4)}{20} = -1.175,$$
$$x^2 = (2.5,\ -1.175,\ 3,\ 4).$$
$$x_3^3 = \frac{53 - (-2 \times 2.5 + 1 \times (-1.175) + 0 \times 4)}{30} = 1.9725,$$
$$x^3 = (2.5,\ -1.175,\ 1.9725,\ 4).$$
$$x_4^4 = \frac{47 - (1 \times 2.5 + 2 \times (-1.175) + 3 \times 1.9725)}{20} = 2.0466,$$
$$x^4 = (2.5,\ -1.175,\ 1.9725,\ 2.0466).$$

Una vez que se ha hecho una iteración completa (n subiteraciones), se utiliza el último x obtenido como aproximación inicial y se vuelve a empezar; se calcula x_1 de tal manera que se satisfaga la primera ecuación, luego se calcula x_2... A continuación están las iteraciones siguientes para el ejemplo anterior.

3.0323	−1.1750	1.9725	2.0466
3.0323	−1.0114	1.9725	2.0466
3.0323	−1.0114	2.0025	2.0466
3.0323	−1.0114	2.0025	1.9991
3.0025	−1.0114	2.0025	1.9991
3.0025	−0.9997	2.0025	1.9991
3.0025	−0.9997	2.0002	1.9991
3.0025	−0.9997	2.0002	1.9998
3.0000	−0.9997	2.0002	1.9998
3.0000	−1.0000	2.0002	1.9998
3.0000	−1.0000	2.0000	1.9998
3.0000	−1.0000	2.0000	2.0000
3.0000	−1.0000	2.0000	2.0000
3.0000	−1.0000	2.0000	2.0000
3.0000	−1.0000	2.0000	2.0000
3.0000	−1.0000	2.0000	2.0000

Teóricamente, el método de Gauss-Seidel puede ser un proceso infinito. En la práctica el proceso se acaba cuando de x^k a x^{k+n} los cambios son muy pequeños. Esto quiere decir que el x actual es casi la solución x^*.

Como el método no siempre converge, entonces otra detención del proceso, no deseada pero posible, está determinada cuando el número de iteraciones realizadas es igual a un número máximo de iteraciones previsto.

El siguiente ejemplo no es convergente, ni siquiera empezando de una aproximación inicial muy cercana a la solución. La solución exacta es $x = (1, 1, 1)$.

Ejemplo 3.2. Resolver

$$\begin{bmatrix} -1 & 2 & 10 \\ 11 & -1 & 2 \\ 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 12 \\ 8 \end{bmatrix}$$

partiendo de $x^0 = (1.0001,\ 1.0001,\ 1.0001)$.

$$\begin{array}{ccc} 1.0012 & 1.0001 & 1.0001 \\ 1.0012 & 1.0134 & 1.0001 \\ 1.0012 & 1.0134 & 0.9660 \end{array}$$

$$\begin{array}{ccc} 0.6863 & 1.0134 & 0.9660 \\ 0.6863 & -2.5189 & 0.9660 \\ 0.6863 & -2.5189 & 9.9541 \end{array}$$

$$\begin{array}{ccc} 83.5031 & -2.5189 & 9.9541 \\ 83.5031 & 926.4428 & 9.9541 \\ 83.5031 & 926.4428 & -2353.8586 \end{array}$$

Algunos criterios garantizan la convergencia del método de Gauss-Seidel. Por ser condiciones suficientes para la convergencia son criterios demasiado fuertes, es decir, la matriz A puede no cumplir estos requisitos y sin embargo el método puede ser convergente. En la práctica, con frecuencia, es muy dispendioso poder aplicar estos criterios.

Una matriz cuadrada es de *diagonal estrictamente dominante por filas* si en cada fila el valor absoluto del elemento diagonal es mayor que la suma de los valores absolutos de los otros elementos de la fila,

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|, \ \ \forall i.$$

Teorema 3.1. *Si A es de diagonal estrictamente dominante por filas, entonces el método de Gauss-Seidel converge para cualquier x^0 inicial.*

Teorema 3.2. *Si A es definida positiva, entonces el método de Gauss-Seidel converge para cualquier x^0 inicial.*

Teóricamente el método de Gauss-Seidel se debería detener cuando $\|x^k - x^*\| < \varepsilon$. Sin embargo la condición anterior necesita conocer x^*, que es precisamente lo que se está buscando. Entonces, de manera práctica el método de GS se detiene cuando $\|x^k - x^{k+n}\| < \varepsilon$.

Dejando de lado los superíndices, las fórmulas del método de Gauss-Seidel se pueden reescribir para facilitar el algoritmo y para mostrar que $\|x^k - x^*\|$

y $\|x^k - x^{k+n}\|$ están relacionadas.

$$\begin{aligned} x_i &\leftarrow \frac{b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j}{a_{ii}}, \\ x_i &\leftarrow \frac{b_i - \sum_{j=1}^{n} a_{ij}x_j + a_{ii}x_i}{a_{ii}}, \\ x_i &\leftarrow x_i + \frac{b_i - A_{i\cdot}\, x}{a_{ii}}. \end{aligned}$$

Sean

$$\begin{aligned} r_i &= b_i - A_{i\cdot}\, x, \\ \delta_i &= \frac{r_i}{a_{ii}}. \end{aligned}$$

El valor r_i es simplemente el error, residuo o resto que se comete en la i-ésima ecuación al utilizar el x actual. Si $r_i = 0$, entonces la ecuación i-ésima se satisface perfectamente. El valor δ_i es la modificación que sufre x_i en una iteración.

Sean $r = (r_1, r_2, ..., r_n)$, $\delta = (\delta_1, \delta_2, ..., \delta_n)$. Entonces $x^{k+n} = x^k + \delta$. Además x^k es solución si y solamente si $r = 0$, o sea, si y solamente $\delta = 0$. Lo anterior justifica que el método de GS se detenga cuando $\|\delta\| \leq \varepsilon$. La norma $\|\delta\|$ puede ser la norma euclidiana o cualquier otra norma.

Si en el criterio de parada del algoritmo se desea enfatizar sobre los errores o residuos, entonces se puede comparar $\|\delta\|$ con $\varepsilon / \|(a_{11}, ..., a_{nn})\|$; por ejemplo,

$$\|\delta\| \leq \frac{\varepsilon}{\max |a_{ii}|}.$$

El esquema del algoritmo para resolver un sistema de ecuaciones por el método de Gauss-Seidel es:

datos: $A,\ b,\ x^0,\ \varepsilon,$ maxit
$x = x^0$
para $k = 1, ...,$maxit
    nrmD$\leftarrow 0$
    para $i = 1, ..., n$
        $\delta_i = (b_i - A_i.\, x)/a_{ii}$
        $x_i \leftarrow x_i + \delta_i$
        nrmd$\leftarrow$nrmD$+|\delta_i|$
    fin-para i
    si nrmD $\leq \varepsilon$ **ent** $x^* \approx x$, **salir**
fin-para k

A continuación hay una versión, no muy eficiente, que permite mostrar los resultados intermedios

```
function [x, ind, k] = GS(A, b, x0, eps, maxit)
  //
  // metodo de Gauss Seidel para resolver  A x = b
  //
  // A   matriz cuadrada,
  // b   vector columna de terminos independientes,
  // x0  vector columna inicial
  //
  // ind valdra -1  si hay un elemento diagonal nulo o casi,
  //
  //             1  si se obtuvo un aproximacion
  //                de la solucion, con la precision deseada,
  //
  //             0  si no se obtuvo una buena aproximacion.
  //
  // k  indicara el numero de iteraciones

  if min( abs(diag(A)) ) <= %eps
    ind = -1
    x = []
    return
  end
  x = x0
  n = size(x,1)
  ind = 1
```

```
    for k = 1:maxit
      //printf('\n k = %d\n', k)
      D = 0
      for i = 1:n
        di = ( b(i) - A(i,:)*x )/A(i,i)
        x(i) = x(i) + di
        D = max(D, abs(di))
      end
      disp(x')
      if D < eps, return, end
    end
    ind = 0
  endfunction
```

En una implementación eficiente para matrices dispersas, se requiere una estructura en la que se almacenan únicamente los elementos no nulos y que permita efectuar el producto de una fila de A por un vector, es decir, que permita remplazar eficientemente la orden `A(i,:)*x`.

3.2 Normas vectoriales

El concepto de norma corresponde simplemente a la abstracción del concepto de tamaño de un vector. Consideremos el vector que va de $(0, 0, 0)$ a $(2, 3, -4)$. Su tamaño o magnitud es simplemente

$$\sqrt{2^2 + 3^2 + (-4)^2} = \sqrt{29}$$

Sea V un espacio vectorial real. Una *norma* es una función

$$\begin{aligned}
&\mu : V \to \mathbb{R} \\
&\mu(x) \geq 0, \quad \forall x \in V, \\
&\mu(x) = 0 \quad \text{sssi} \quad x = 0, \\
&\mu(\alpha x) = |\alpha|\, \mu(x), \quad \forall \alpha \in \mathbb{R}, \ \forall x \in V, \\
&\mu(x + y) \leq \mu(x) + \mu(y), \ \forall x, y \ \in V. \quad \text{(desigualdad triangular)}
\end{aligned}$$

Ejemplos clásicos de normas en $\mathbb{R}^n$ son:

$$||x||_2 = ||x|| = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2} \quad \text{norma euclidiana,}$$

$$||x||_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p} \quad \text{norma de Holder de orden } p \geq 1,$$

$$||x||_\infty = ||x||_{\max} = \max_{1 \leq i \leq n} |x_i|,$$

$$\alpha||x|| \quad \text{con } \alpha > 0 \quad \text{y} \quad || \quad || \text{ una norma,}$$

$$||x||_A = \sqrt{x^{\mathrm{T}} A x} \quad \text{con } A \text{ definida positiva.}$$

Se puede mostrar que

$$\lim_{p \to \infty} ||x||_p = ||x||_\infty = ||x||_{\max}.$$

Sea $x = (3, 0, -4)$, entonces

$$\begin{aligned} ||x||_1 &= 7, \\ ||x||_2 &= 5, \\ ||x||_\infty &= 4. \end{aligned}$$

3.2.1 En Scilab

Si `x` es un vector fila o columna, entonces

`norm(x)`	calcula $\|\|x\|\|_2$,
`norm(x, 2)`	calcula $\|\|x\|\|_2$,
`norm(x, 1)`	calcula $\|\|x\|\|_1$,
`norm(x, 4)`	calcula $\|\|x\|\|_4$,
`norm(x, 'inf')`	calcula $\|\|x\|\|_\infty$.

3.3 Normas matriciales

En el conjunto de matrices cuadradas de orden n se puede utilizar cualquier norma definida sobre $\mathbb{R}^{n^2}$. Dado que en el conjunto de matrices cuadradas

está definido el producto, es interesante contar con normas que tengan características especiales relativas al producto entre matrices y al producto entre una matriz y un vector. En particular en algunos casos es conveniente que se tengan estas dos propiedades:

$$||AB|| \leq ||A||\ ||B||,$$
$$||Ax|| \leq ||A||\ ||x||.$$

Ejemplo 3.3. Sean

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}, \quad x = \begin{bmatrix} 5 \\ 6 \end{bmatrix},$$

entonces

$$AB = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}, \quad Ax = \begin{bmatrix} 17 \\ 39 \end{bmatrix},$$

pero

$$||AB||_\infty = 50, \qquad ||A||_\infty ||B||_\infty = 4 \times 8 = 32$$
$$||Ax||_\infty = 39, \qquad ||A||_\infty ||x||_\infty = 4 \times 6 = 24. \ \diamond$$

Una norma $||\ \ ||$ definida sobre el $\mathbb{R}^{n\times n}$ (conjunto de matrices $n \times n$) se llama *matricial* o (submultiplicativa) si, además de las propiedades usuales de norma, para cualquier par de matrices A y B

$$||AB|| \leq ||A||\,||B||.$$

Sean $||\ \ ||_m$ una norma matricial sobre $\mathbb{R}^{n\times n}$ y $||\ \ ||_v$ una norma sobre $\mathbb{R}^n$. Estas dos normas se llaman *compatibles* o *consistentes* si, para toda matriz $A \in \mathbb{R}^{n\times n}$ y para todo $x \in \mathbb{R}^n$

$$||Ax||_v \leq ||A||_m ||x||_v\,.$$

Una manera común de construir normas que sean matriciales y compatibles es generando una norma a partir de un norma sobre $\mathbb{R}^n$. Sea $||\ \ ||$ una norma

sobre $\mathbb{R}^n$. La *norma generado o inducida* por esta norma se define de varias maneras, todas ellas equivalentes:

$$|||A||| = \sup_{x\neq 0} \frac{||Ax||}{||x||} \tag{3.1}$$

$$|||A||| = \max_{x\neq 0} \frac{||Ax||}{||x||} \tag{3.2}$$

$$|||A||| = \sup_{||x||=1} ||Ax|| \tag{3.3}$$

$$|||A||| = \max_{||x||=1} ||Ax||. \tag{3.4}$$

Proposición 3.1. *La definición anterior está bien hecha, es decir,* $|||\ \ |||$ *es una norma, es matricial y es compatible con* $||\ \ ||$.

Demostración. Sea

$$\mu(A) = \sup_{x\neq 0} \frac{||Ax||}{||x||}$$

Ante todo es necesario mostrar que la función μ está bien definida, o sea, para toda matriz A,

$$\mu(A) = \sup_{x\neq 0} \frac{||Ax||}{||x||} < \infty.$$

Veamos

$$\begin{aligned}
\mu(A) &= \sup_{x\neq 0} \frac{\left|\left|A\Big(||x||\frac{x}{||x||}\Big)\right|\right|}{||x||} \\
&= \sup_{x\neq 0} \frac{\left|\left|\,||x||A\frac{x}{||x||}\,\right|\right|}{||x||} \\
&= \sup_{x\neq 0} \frac{||x||\left|\left|A\frac{x}{||x||}\right|\right|}{||x||} \\
&= \sup_{x\neq 0} \left|\left|A\frac{x}{||x||}\right|\right| \\
&= \sup_{||\xi||=1} ||A\xi||
\end{aligned}$$

La función $\xi \mapsto \varphi(\xi) = ||A\xi||$ es continua y el conjunto $S = \{\xi \in \mathbb{R}^n : ||\xi|| = 1\}$ es compacto (cerrado y acotado), luego $\varphi(S)$ es compacto, en particular acotado, es decir, $\mu(A) = \sup \varphi(S) < \infty$. Además el sup se alcanza en un punto de S. Luego las 4 definiciones, (3.1) y siguientes, coinciden.

Claramente $\mu(A) \geq 0$. Veamos que $\mu(A) = 0$ sssi $A = 0$. Si $A = 0$, entonces $\mu(A) = 0$. Sea $A \neq 0$. Entonces A tiene por lo menos una columna no nula. Sea $A_{\cdot j} \neq 0$ y $v = e^j/||e^j||$. Por definición $||v|| = 1$.

$$\begin{aligned}
\mu(A) \geq ||Av|| &= \left|\left|A\frac{e^j}{||e^j||}\right|\right| \\
&= \left|\left|\frac{A_{\cdot j}}{||e^j||}\right|\right| \\
&= \frac{||A_{\cdot j}||}{||e^j||} > 0.
\end{aligned}$$

$$\mu(\lambda A) = \max_{||x||=1} ||\lambda Ax|| = \max_{||x||=1} |\lambda|\ ||Ax|| = |\lambda| \max_{||x||=1} ||Ax|| = |\lambda|\mu(A).$$

Para mostrar que $\mu(A + B) \leq (A) + \mu(B)$ se usa la siguiente propiedad:

$$\sup_{x\in X}(\ f(x)+g(x)\ ) \le \sup_{x\in X} f(x) + \sup_{x\in X} g(x)$$

$$\mu(A+B) = \sup_{||x||=1} ||(A+B)x|| = \sup_{||x||=1} ||Ax+Bx|| \le \sup_{||x||=1} (||Ax||+||Bx||)$$
$$\le \sup_{||x||=1} ||Ax|| + \sup_{||x||=1} ||Bx|| = \mu(A)+\mu(B)$$

Hasta ahora se ha mostrado que μ es una norma sobre $\mathbb{R}^{n\times n}$. Si se utilizó la norma $||\ \ ||_\square$ en $\mathbb{R}^n$, la norma *generada* o *subordinada* sobre $\mathbb{R}^{n\times n}$ se denota por $|||\ \ |||_\square$. **Cuando no hay ambigüedad, es la notación más usual,** $||A||_\square$ **indica la norma generada evaluada en la matriz** A **y** $||x||_\square$ **indica la norma original evaluada en el vector columna** x**.**

Veamos ahora que la norma original y la generada son compatibles. Obviamente si $x = 0$, entonces $||Ax|| \le ||A||\ ||x||$. Sea $x \neq 0$ y $\xi = x/||x||$ de norma uno.

$$||A|| \ge ||A\xi|| = \left|\left|A\frac{x}{||x||}\right|\right| = \frac{||Ax||}{||x||}, \quad \text{luego} \quad ||A||\ ||x|| \ge ||Ax||.$$

Queda por mostrar que esta norma generada es matricial.

$$||AB|| = \max_{||x||=1} ||ABx|| = \max_{||x||=1} ||A(Bx)|| \le \max_{||x||=1} ||A||\ ||Bx||$$
$$= ||A|| \max_{||x||=1} ||Bx|| = ||A||\ ||B||. \qquad \square$$

Para las 3 normas vectoriales más usadas, las normas matriciales generadas son:

$$||A||_1 = \max_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}|, \tag{3.5}$$

$$||A||_2 = \sqrt{\rho(A^{\mathrm{T}}A)}\ \text{(norma espectral)}, \tag{3.6}$$

$$||A||_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|. \tag{3.7}$$

Si la matriz A se considera como un vector, entonces se puede aplicar la norma euclidiana. Esta norma resulta ser matricial. Esta norma se conoce con el nombre de norma de Frobenius o también de Schur.

$$||A||_F = \left(\sum_{i,j} (a_{ij})^2 \right)^{1/2} . \tag{3.8}$$

Para cualquier norma generada $||I|| = 1$. Como $||I||_F = \sqrt{n}$, entonces esta norma no puede ser generada por ninguna norma vectorial

Ejemplo 3.4. Sea

$$A = \begin{bmatrix} 1 & 2 \\ 3 & -4 \end{bmatrix}$$

Entonces

$$A^{\mathrm{T}} A = \begin{bmatrix} 10 & -10 \\ -10 & 20 \end{bmatrix}$$

Sus valores propios son 3.8196601 y 26.18034. Luego

$$\begin{aligned} ||A||_1 &= 6, \\ ||A||_2 &= 5.1166727, \\ ||A||_\infty &= 7. \end{aligned}$$

Proposición 3.2. . $||A||_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$

Demostración.

$$\begin{aligned}
||A||_1 &= \max_{||x||_1=1} ||Ax||_1 \\
&= \max_{||x||_1=1} \sum_{i=1}^{n} |(Ax)_i| \\
&= \max_{||x||_1=1} \sum_{i=1}^{n} |A_i.x| \\
&= \max_{||x||_1=1} \sum_{i=1}^{n} \left| \sum_{j=1}^{n} a_{ij}x_j \right| \\
&\leq \max_{||x||_1=1} \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}x_j| \\
&= \max_{||x||_1=1} \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|\,|x_j| \\
&= \max_{||x||_1=1} \sum_{j=1}^{n} |x_j| \sum_{i=1}^{n} |a_{ij}| \\
&= \max_{||x||_1=1} \sum_{j=1}^{n} |x_j| s_j
\end{aligned}$$

donde $s_j = \sum_{i=1}^{n} |a_{ij}|$. Si $\alpha_j, \beta_j \geq 0$ para todo j, entonces

$$\sum_{j=1}^{n} \alpha_j \beta_j \leq \left(\max_j \beta_j \right) \left(\sum_{j=1}^{n} \alpha_j \right).$$

Luego

$$
\begin{aligned}
||A||_1 &\leq \max_{||x||_1=1} \left(\left(\max_j s_j \right) \left(\sum_{j=1}^{n} |x_j| \right) \right) \\
&= \max_{||x||_1=1} \left(\max_j s_j \right) \\
&= \max_j s_j \\
&= \max_j \sum_{i=1}^{n} |a_{ij}|
\end{aligned}
$$

En resumen

$$||A||_1 \leq \max_j \sum_{i=1}^{n} |a_{ij}|.$$

Sea k tal que

$$\sum_{i=1}^{n} |a_{ik}| = \max_j \sum_{i=1}^{n} |a_{ij}|$$

$$
\begin{aligned}
||A||_1 &= \max_{||x||_1=1} ||Ax||_1 \\
&\geq ||Ax||_1 \quad \text{para todo } x \text{ con } ||x||_1 = 1 \\
||A||_1 &\geq ||Ae^k||_1 \\
&= ||A_{\cdot k}||_1 \\
&= \sum_{i=1}^{n} |a_{ik}| = \max_j \sum_{i=1}^{n} |a_{ij}|
\end{aligned}
$$

es decir,

$$||A||_1 \geq \max_j \sum_{i=1}^{n} |a_{ij}|. \quad \square$$

Proposición 3.3. $||A||_2 = \sqrt{\rho(A^T A)}$.

Demostración.

$$
\begin{aligned}
||A||_2 &= \max_{||x||_2=1} ||Ax||_2 \\
||A||_2^2 &= \max_{||x||_2=1} ||Ax||_2^2 \\
||A||_2^2 &= \max_{||x||_2=1} x^{\mathrm{T}} A^{\mathrm{T}} Ax
\end{aligned}
$$

La matriz $A^{\mathrm{T}}A$ es simétrica y semidefinida positiva, todos sus valores propios λ_1, ..., λ_n son reales y no negativos. Sea

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0.$$

Por el teorema espectral, $A^{\mathrm{T}}A$ es semejante, ortogonalmente, a la matriz diagonal de sus valores propios. Las matrices se pueden reordenar para que

$$V^{\mathrm{T}}(A^{\mathrm{T}}A)V = \mathrm{diag}(\lambda_1, ..., \lambda_n)\,, \quad \text{con } V \text{ ortogonal.}$$

Sean v^1, v^2, ..., v^n las columnas de V. Entonces v^1, v^2, ..., v^n forman un conjunto ortonormal de vectores propios, es decir,

$$
\begin{aligned}
(A^{\mathrm{T}}A)v^i &= \lambda_i v^i, \\
{v^i}^{\mathrm{T}} v^j &= \delta_{ij}\,.
\end{aligned}
$$

Sea x tal que $||x||_2 = 1$, $\alpha = V^{\mathrm{T}}x$. Entonces $||\alpha||_2 = 1$ y $V\alpha = VV^{\mathrm{T}}x = x$, es decir,

$$x = \sum_{i=1}^{n} \alpha_i v^i.$$

Entonces

$$
\begin{aligned}
A^{\mathrm{T}}Ax &= A^{\mathrm{T}}A\sum_{i=1}^{n}\alpha_i v^i \\
&= \sum_{i=1}^{n}\alpha_i A^{\mathrm{T}}Av^i \\
&= \sum_{i=1}^{n}\alpha_i \lambda_i v^i \\
x^{\mathrm{T}}A^{\mathrm{T}}Ax &= \left(\sum_{j=1}^{n}\alpha_j v^j\right)^{\mathrm{T}}\left(\sum_{i=1}^{n}\alpha_i\lambda_i v^i\right) \\
&= \sum_{i=1}^{n}\alpha_i^2\lambda_i \\
&\leq \lambda_1\sum_{i=1}^{n}\alpha_i^2 \\
&= \lambda_1
\end{aligned}
$$

En resumen,

$$
\begin{aligned}
||A||_2^2 &\leq \lambda_1 \\
||A||_2 &\leq \sqrt{\lambda_1}
\end{aligned}
$$

Por otro lado,

$$
\begin{aligned}
&||A||_2 \geq \sqrt{x^{\mathrm{T}}A^{\mathrm{T}}Ax} \quad \text{pra todo } x \text{ con } ||x||_2 = 1 \\
&||A||_2 \geq \sqrt{v^{1\mathrm{T}}A^{\mathrm{T}}Av^1} \\
&||A||_2 \geq \sqrt{v^{1\mathrm{T}}\lambda_1 v^1} \\
&||A||_2 \geq \sqrt{\lambda_1 v^{1\mathrm{T}}v^1} \\
&||A||_2 \geq \sqrt{\lambda_1}. \qquad \square
\end{aligned}
$$

Proposición 3.4. $||A||_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$

Demostración.

$$\begin{aligned}
||A||_\infty &= \max_{||x||_\infty=1} ||Ax||_\infty \\
&= \max_{||x||_\infty=1} \max_i |(Ax)_i| \\
&= \max_{||x||_\infty=1} \max_i |A_{i\cdot}x| \\
&= \max_{||x||_\infty=1} \max_i |\sum_{j=1}^n a_{ij}x_j| \\
&\leq \max_{||x||_\infty=1} \max_i \sum_{j=1}^n |a_{ij}|\,|x_j|
\end{aligned}$$

Como $|x_j| \leq ||x||_\infty$

$$\begin{aligned}
||A||_\infty &\leq \max_{||x||_\infty=1} \max_i \sum_{j=1}^n |a_{ij}|\,||x||_\infty \\
&= \max_{||x||_\infty=1} ||x||_\infty \max_i \sum_{j=1}^n |a_{ij}| \\
&= \max_i \sum_{j=1}^n |a_{ij}|
\end{aligned}$$

Veamos ahora la otra desigualdad. Si $A = 0$, se cumple la igualdad. Sean k y $\bar{x}$ tales que

$$\sum_{j=1}^n |a_{kj}| = \max_i \sum_{j=1}^n |a_{ij}|$$

$$\bar{x}_j = \begin{cases} 0 & \text{si } a_{kj} = 0 \\ \text{signo}(a_{kj}) = \dfrac{|a_{kj}|}{a_{kj}} & \text{si } a_{kj} \neq 0. \end{cases}$$

$$
\begin{aligned}
&||x||_\infty = 1, \\
&||A||_\infty \geq ||Ax||_\infty \quad \text{si} \quad ||x||_\infty = 1, \\
&||A||_\infty \geq ||A\bar{x}||_\infty \\
&\qquad = \max_i |A_{i\cdot}\bar{x}| \\
&\qquad = |A_{i\cdot}\bar{x}| \quad \text{para todo } i, \\
&||A||_\infty \geq |A_{k\cdot}\bar{x}| \\
&\qquad = \left| \sum_{j=1}^{n} a_{kj} \frac{|a_{kj}|}{a_{kj}} \right| \\
&\qquad = \left| \sum_{j=1}^{n} |a_{kj}| \right| \\
&\qquad = \sum_{j=1}^{n} |a_{kj}| \\
&\qquad = \max_i \sum_{j=1}^{n} |a_{ij}|.
\end{aligned}
$$

En las sumas de las desigualdades anteriores, los términos donde $a_{kj} = 0$ no se consideran. □

Proposición 3.5. . *Si* $||\ \ ||_\alpha$ *es una norma matricial, entonces existe por lo menos una norma vectorial compatible con ella.*

Demostración. Sean $X = [x\ 0\ 0\ \cdots\ 0] \in \mathbb{R}^{n\times n}$ y $||x|| = ||X||_\alpha$. Se puede comprobar que $||\ \ ||$ es una norma en $\mathbb{R}^n$ y que es compatible con $||\ \ ||_\alpha$.

RESUMEN DE RESULTADOS

- $||\ \ ||$ (definida en (3.4)) es una norma.
- $||\ \ ||$ (definida en (3.4)) es matricial.
- $||\ \ ||$ (para matrices) es compatible con $||\ \ ||$ (para vectores columna).
- $||I|| = 1$.

- $||A||_1 = \max_j \sum_{i=1}^{n} |a_{ij}|$
- $||A||_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$
- $||A||_2 = \sqrt{\rho(A^{\mathrm{T}}A)}$
- $||A||_2 = \max\{\sigma_1, \sigma_2, ..., \sigma_n\} = \max\{\text{valores singulares de } A\}$ (ver [AlK02]).
- $||A||_2 = \rho(A)$ si $A \succeq 0$.
- $||A||_F = \sqrt{\mathrm{tr}(A^{\mathrm{T}}A)} = \sqrt{\sum_{i=1}^{n} \sigma_i^2}$
- Si Q es ortogonal $||QA||_F = ||AQ||_F = ||A||_F$.
- $||A||_2 \leq ||A||_F \leq \sqrt{n}||A||_2$
- $||A||_2 = ||A||_F$ sssi $r(A) = 1$.
- $\frac{1}{\sqrt{n}}||A||_1 \leq ||A||_F \leq \sqrt{n}||A||_1$
- $\frac{1}{\sqrt{n}}||A||_\infty \leq ||A||_F \leq \sqrt{n}||A||_\infty$
- $||A||_2^2 \leq ||A||_1||A||_\infty$
- $\rho(A) \leq ||A||$ para toda norma matricial $||\ \ ||$.
- Sea $\varepsilon > 0$. Entonces existe una norma matricial $||\ \ ||$ tal que
$$||A|| \leq \rho(A) + \varepsilon$$
- $||\ \ ||_F$ es multiplicativa (ver [Ste98]).
- $||\ \ ||_F$ y $||\ \ ||_2$ son compatibles.
- $||\ \ ||_F$ no es la norma generada por ninguna norma $||\ \ ||$ ya que $||I||_F = \sqrt{n} \neq 1$.
- $n \max_{i,j} |a_{ij}|$ es matricial (ver [Man04]).
- $n \max_{i,j} |a_{ij}|$ es compatible con $||\ \ ||_1$, $||\ \ ||_2$ y $||\ \ ||_\infty$.

3.3.1 En Scilab

Si `A` es una matriz, entonces

`norm(A)`	calcula $\|\|A\|\|_2$,
`norm(A, 2)`	calcula $\|\|A\|\|_2$,
`norm(A, 1)`	calcula $\|\|A\|\|_1$,
`norm(A, 'inf')`	calcula $\|\|A\|\|_\infty$,
`norm(A, 'fro')`	calcula $\|\|A\|\|_F$.

3.4 Condicionamiento de una matriz

Cuando se resuelve un sistema de ecuaciones $Ax = b$ se desea conocer cómo son los cambios en la solución cuando se cambia ligeramente el vector de términos independientes b.

De manera más precisa, sea $\bar{x}$ la solución de $Ax = b$ y $\bar{x}'$ la solución de $Ax = b'$. Se puede suponer que

$$\begin{aligned} b' &= b + \Delta b, \\ \bar{x}' &= \bar{x} + \Delta x. \end{aligned}$$

Se espera que si $||\Delta b||$ es pequeña, entonces también $||\Delta x||$ es pequeña. En realidad es mejor considerar cambios relativos. Se espera que si el valor $||\Delta b||/||b||$ es pequeño, entonces también $||\Delta x||/||\bar{x}||$ sea pequeño. Las deducciones que siguen relacionan los dos cambios relativos.

$$\begin{aligned} \Delta x &= \bar{x}' - \bar{x} \\ &= A^{-1}b' - A^{-1}b \\ &= A^{-1}(b + \Delta b) - A^{-1}b \\ &= A^{-1}\Delta b. \end{aligned}$$

Al utilizar una norma y la norma matricial generada se obtiene

$$||\Delta x|| \leq ||A^{-1}||\ ||\Delta b||.$$

Por otro lado

$$\begin{aligned} b &= Ax \\ ||b|| &\leq ||A||\,||\bar{x}|| \\ \frac{1}{||\bar{x}||} &\leq \frac{||A||}{||b||} \end{aligned}$$

Multiplicando la primera y la última desigualdad

$$\frac{||\Delta x||}{||\bar{x}||} \leq ||A||\ ||A^{-1}||\frac{||\Delta b||}{||b||}. \tag{3.9}$$

El valor $||A||\ ||A^{-1}||$ se llama condicionamiento o número de condición de la matriz A (invertible) y se denota

$$\kappa(A) = ||A||\ ||A^{-1}||.$$

Entonces

$$\frac{||\Delta x||}{||\bar{x}||} \leq \kappa(A)\frac{||\Delta b||}{||b||}. \tag{3.10}$$

Ejemplo 3.5. Calcular $\kappa_1(A)$, $\kappa_2(A)$ y $\kappa_\infty(A)$ para la matriz

$$A = \begin{bmatrix} -10 & -7 \\ 6 & 4 \end{bmatrix}.$$

Entonces

$$\begin{aligned}
A^{-1} &= \begin{bmatrix} 2 & 7/2 \\ -3 & -5 \end{bmatrix} \\
A^{\mathrm{T}}A &= \begin{bmatrix} 136 & 94 \\ 94 & 65 \end{bmatrix} \\
A^{-1^{\mathrm{T}}}A^{-1} &= \begin{bmatrix} 13 & 22 \\ 22 & 149/4 \end{bmatrix} \\
\mathrm{esp}(A^{\mathrm{T}}A) &= \{0.0199025, 200.9801\} \\
\mathrm{esp}(A^{-1^{\mathrm{T}}}A^{-1}) &= \{0.0049756, 50.245024\} \\
||A||_2 &= 14.176745 \\
||A^{-1}||_2 &= 7.0883725 \\
\kappa_2(A) &= 100.49005 \\
||A||_1 &= 16 \\
||A^{-1}||_1 &= 17/2 \\
\kappa_1(A) &= 136 \\
||A||_\infty &= 17 \\
||A^{-1}||_\infty &= 8 \\
\kappa_\infty(A) &= 136. \ \diamond
\end{aligned}$$

El condicionamiento, definido para normas matriciales inducidas de normas vectoriales, tiene la siguientes propiedades:

- $\kappa(A) \geq 1$.
- $\kappa(\alpha A) = \kappa(A)$ si $\alpha \neq 0$.
- $\kappa_2(A) = 1$ si y solamente si A es un múltiplo de una matriz ortogonal (o unitaria).

La desigualdad (3.10) indica que si $\kappa(A)$ es pequeño, entonces un cambio relativo en b pequeño produce un cambio relativo en x pequeño.

Una matriz A es bien condicionada si $\kappa(A)$ es cercano a 1 y es mal condicionada si $\kappa(A)$ es grande. Para el condicionamiento κ_2 (definido con la norma espectral) las matrices mejor condicionadas son las matrices ortogonales.

Ejemplo 3.6. Resolver los sistemas $Ax = b$ y $Ax' = b'$, donde

$$A = \begin{bmatrix} 10 & 10 \\ 10 & -9 \end{bmatrix}, \qquad b = \begin{bmatrix} 20.01 \\ 19.99 \end{bmatrix}, \qquad b' = \begin{bmatrix} 20.02 \\ 19.98 \end{bmatrix}.$$

Entonces

$$\begin{aligned} \Delta b &= [0.01 \quad -0.01]^{\mathrm{T}}, \\ \frac{||\Delta b||}{||b||} &= 0.0005, \\ \kappa(A) &= 1.0752269. \end{aligned}$$

Al resolver los dos sistemas se obtiene:

$$\begin{aligned} x &= [1.9999474 \quad 0.0010526]^{\mathrm{T}}, \\ x' &= [1.9998947 \quad 0.0021053]^{\mathrm{T}}, \\ \Delta x &= [-0.0000526 \quad .0010526]^{\mathrm{T}}, \\ \frac{||\Delta x||}{||x||} &= 0.0005270, \\ \kappa(A)\frac{||\Delta b||}{||b||} &= 0.0005376. \end{aligned}$$

La matriz A es muy bien condicionada y entonces cambios pequeños en b producen cambios pequeños en x. $\diamond$

Ejemplo 3.7. Resolver los sistemas $Ax = b$ y $Ax' = b'$, donde

$$A = \begin{bmatrix} 10.01 & 10.00 \\ 10.00 & 9.99 \end{bmatrix}, \qquad b = \begin{bmatrix} 20.01 \\ 19.99 \end{bmatrix}, \qquad b' = \begin{bmatrix} 20.02 \\ 19.98 \end{bmatrix}.$$

Entonces

$$\begin{aligned} \Delta b &= [0.01 \quad -0.01]^{\mathrm{T}}, \\ \frac{||\Delta b||}{||b||} &= 0.0005, \\ A^{-1} &= \begin{bmatrix} -99900 & 100000 \\ 100000 & -100100 \end{bmatrix}, \\ \kappa(A) &= 4000002. \end{aligned}$$

Al resolver los dos sistemas se obtiene:

$$\begin{aligned} x &= [1 \quad 1]^{\mathrm{T}}, \\ x' &= [-1998 \quad 2002]^{\mathrm{T}}, \\ \Delta x &= [-1999 \quad 2001]^{\mathrm{T}}, \\ \frac{||\Delta x||}{||x||} &= 2000.0002, \\ \kappa(A)\frac{||\Delta b||}{||b||} &= 2000.0008. \end{aligned}$$

La matriz A es muy mal condicionada y entonces cambios pequeños en b pueden producir cambios muy grandes en la solución. $\diamond$

Ejemplo 3.8. Resolver los sistemas $Ax = b$ y $Ax'' = b''$, donde

$$A = \begin{bmatrix} 10.01 & 10.00 \\ 10.00 & 9.99 \end{bmatrix}, \qquad b = \begin{bmatrix} 20.01 \\ 19.99 \end{bmatrix}, \qquad b'' = \begin{bmatrix} 20.02 \\ 20.00 \end{bmatrix}.$$

Entonces

$$\begin{aligned} \Delta b &= [0.01 \quad 0.01]^{\mathrm{T}}, \\ \frac{||\Delta b||}{||b||} &= 0.0005, \\ A^{-1} &= \begin{bmatrix} -99900 & 100000 \\ 100000 & -100100 \end{bmatrix}, \\ \kappa(A) &= 4000002. \end{aligned}$$

Al resolver los dos sistemas se obtiene:

$$\begin{aligned} x &= [1 \quad 1]^{\mathrm{T}}, \\ x'' &= [2 \quad 0]^{\mathrm{T}}, \\ \Delta x &= [1 \quad -1]^{\mathrm{T}}, \\ \frac{||\Delta x||}{||x||} &= 1, \\ \kappa(A)\frac{||\Delta b||}{||b||} &= 2000.0008. \end{aligned}$$

La matriz A, la misma del ejemplo anterior, es muy mal condicionada y entonces cambios pequeños en b pueden producir cambios muy grandes en la solución. Sin embargo los cambios en la solución, aunque no despreciables, no fueron tan grandes como en el ejemplo anterior, o sea, $||\Delta x||/||x||$ está lejos de la cota superior. $\diamond$

En Scilab el condicionamiento para la norma euclidiana se calcula por medio de `cond( A )`.

3.5 Método de Jacobi

Este método se parece al método GS, también se utiliza la ecuación i-ésima para calcular x_i y el cálculo de x_i se hace de la misma forma. Pero un valor recién calculado de x_i no se utiliza inmediatamente. Los valores nuevos de x_i solamente se empiezan a utilizar cuando ya se calcularon todos los n valores x_i.

Ejemplo 3.9.

$$A = \begin{bmatrix} 4 & 1 & -1 \\ 2 & 5 & 0 \\ -2 & 3 & 10 \end{bmatrix}, \quad b = \begin{bmatrix} 7 \\ 19 \\ 45 \end{bmatrix}, \quad x^0 = \begin{bmatrix} 1.2 \\ 1.5 \\ 1.6 \end{bmatrix}.$$

Gauss-Seidel			Jacobi		
x_1	x_2	x_3	x_1	x_2	x_3
1.2	1.5	1.6	1.2	1.5	1.6
1.775	1.5	1.6	1.775	1.5	1.6
1.775	3.09	1.6	1.775	3.32	1.6
1.775	3.09	3.928	1.775	3.32	4.29
1.9595	3.09	3.928	1.9925	3.32	4.29
1.9595	3.0162	3.928	1.9925	3.09	4.29
1.9595	3.0162	3.98704	1.9925	3.09	3.859

El primer vector calculado es igual en los dos métodos. Para calcular x_2 en el método GS se usa el valor $x_1 = 1.775$ recién calculado:

$$x_2 = \frac{19 - 2 \times 1.775 - 0 \times 1.6}{5} = 3.09\,.$$

En cambio en el método de Jacobi:

$$x_2 = \frac{19 - 2 \times 1.2 - 0 \times 1.6}{5} = 3.32\,.$$

En el método de GS:

$$x_3 = \frac{45 + 2 \times 1.775 - 3 \times 3.09}{10} = 3.928\,.$$

En el método de Jacobi:

$$x_3 = \frac{45 + 2 \times 1.2 - 3 \times 1.5}{10} = 4.29\,.$$

Ahora sí, en el método de Jacobi, los valores calculados de x_2 y x_3 se utilizan para volver a clacular x_1. $\diamond$

3.6 Método iterativo general

Muchos método iterativos, en particular, los métodos de Jacobi, GS, SOR se pueden expresar de la forma

$$x^{k+1} = Mx^k + p. \tag{3.11}$$

Al aplicar varias veces la fórmula anterior, se está buscando un punto fijo de la función $f(x) = Mx + p$. Al aplicar el teorema de punto fijo de Banach, uno de los resultados más importantes del análisis matemático, se tiene el siguiente resultado.

Teorema 3.3. *Si existe una norma matricial* $||\ \ ||$ *tal que*

$$||M|| < 1.$$

entonces existe un único punto fijo x^* *tal que* $x^* = Mx^* + p$. *Este punto se puede obtener como límite de la iteración* (3.11) *para cualquier* x^0 *inicial.*

En algunos casos el criterio anterior se puede aplicar fácilmente al encontrar una norma adecuada. Pero por otro lado, si después de ensayar con varias normas, no se ha encontrado una norma que sirva, no se puede concluir que no habrá convergencia. El siguiente criterio es más preciso pero puede ser numéricamente más difícil de calcular.

Teorema 3.4. *La iteración de punto fijo* (3.11) *converge si y solamente si*

$$\rho(M) < 1.$$

El *radio espectral* de una matriz cuadrada M, denotado generalmente $\rho(M)$, es la máxima norma de los valores propios de M (reales o complejos),

$$\rho(M) = \max_{1 \le i \le n} \{|\lambda_i| : \lambda_i \in \text{esp}(M)\},$$

donde esp(M) es el conjunto de valores propios de M.

La convergencia es lenta cuando $\rho(M)$ es cercano a 1, es rápida cuando $\rho(M)$ es pequeño (cercano a 0).

Cualquier matriz cuadrada A se puede expresar de la forma

$$A = L + D + U,$$

donde L es la matriz triangular inferior correspondiente a la parte triangular estrictamente inferior de A, D es la matriz diagonal correspondiente a los elementos diagonales de A y U es la matriz triangular superior correspondiente a la parte triangular estrictamente superior de A.

Para el método de Jacobi:

$$M_{\text{J}} = -D^{-1}(L + U),$$
$$p_{\text{J}} = D^{-1}b.$$

Para el método GS

$$M_{\text{GS}} = -(D + L)^{-1}U,$$
$$p_{\text{GS}} = (D + L)^{-1}b.$$

3.7 Método de sobrerrelajación

Este método, conocido como SOR (Successive Over Relaxation), se puede considerar como una generalización del método GS. La fórmulas que definen el método GS son:

$$r_i = b_i - A_{i}.x\,,$$
$$\delta_i = \frac{r_i}{a_{ii}}\,,$$
$$x_i = x_i + \delta_i\,.$$

El el método SOR únicamente cambia la última asignación, introduciendo un parámetro ω,

$$\begin{aligned} r_i &= b_i - A_i.x\,, \\ \delta_i &= \frac{r_i}{a_{ii}}\,, \\ x_i &= x_i + \omega\delta_i\,. \end{aligned} \tag{3.12}$$

Si $0 < \omega < 1$ se tiene una subrrelajación, si $1 < \omega$ se tiene la sobrerrelajación propiamente dicha. Si $\omega = 1$, se tiene el método GS. Una escogencia adecuada de ω mejora la convergencia del método GS. Este método se usa en algunas técnicas de solución de ecuaciones diferenciales parciales.

Una condición necesaria para que el método SOR converja, ver [Dem97], es que

$$0 < \omega < 2\,.$$

Para matrices definidas positivas el método SOR converge para cualquier ω en el intervalo $]0, 2[$.

Ejemplo 3.10. Resolver el sistema $Ax = b$ por el método SOR con $\omega = 1.4$ partiendo de $x^0 = (1, 1, 1, 1)$.

$$A = \begin{bmatrix} 5 & -1 & 2 & -2 \\ 0 & 4 & 2 & 3 \\ 3 & 3 & 8 & -2 \\ -1 & 4 & -1 & 6 \end{bmatrix}, \qquad b = \begin{bmatrix} 25 \\ -10 \\ 35 \\ -33 \end{bmatrix}.$$

Entonces

$$
\begin{aligned}
r_1 &= b_1 - A_1.x = 25 - 4 = 21 \\
\delta_1 &= \frac{21}{5} = 4.2 \\
\omega\delta_1 &= 5.88 \\
x_1 &= 1 + 5.88 = 6.88
\end{aligned}
$$

$$
\begin{aligned}
r_2 &= -10 - 9 = -19 \\
\delta_2 &= \frac{-19}{4} = -4.75 \\
\omega\delta_2 &= -6.65 \\
x_2 &= 1 - 6.65 = -5.65
\end{aligned}
$$

$$
\begin{aligned}
r_3 &= 35 - 9.69 = 25.31 \\
\delta_3 &= \frac{25.31}{8} = 3.163750 \\
\omega\delta_3 &= 4.429250 \\
x_3 &= 1 + 4.429250 = 5.429250
\end{aligned}
$$

$$
\begin{aligned}
r_4 &= -33 - -28.909250 = -4.090750 \\
\delta_4 &= \frac{-4.090750}{6} = -0.681792 \\
\omega\delta_4 &= -0.954508 \\
x_4 &= 1 - 0.954508 = 0.045492
\end{aligned}
$$

$$
\begin{aligned}
r_1 &= 25 - 50.817517 = -25.817517 \\
\delta_1 &= \frac{-25.817517}{5} = -5.163503 \\
\omega\delta_1 &= -7.228905 \\
x_1 &= 6.880000 + -7.228905 = -0.348905
\end{aligned}
$$

La siguiente tabla muestra las primeras 15 iteraciones completas

Sobrerrelajación, $\omega = 1.4$.

k	x_1	x_2	x_3	x_4
0	1.000000	1.000000	1.000000	1.000000
1	6.880000	-5.650000	5.429250	0.045492
2	-0.348905	-5.088241	6.823724	-1.458380
3	1.076876	-4.710011	4.792473	-1.351123
4	1.810033	-3.552048	4.649676	-2.337041
5	1.368852	-2.880061	4.240550	-2.768266
6	1.721105	-2.409681	3.821389	-3.050409
7	1.788640	-2.008170	3.644054	-3.337915
8	1.812353	-1.742759	3.462571	-3.507443
9	1.883878	-1.543881	3.333868	-3.638593
10	1.909584	-1.395632	3.248121	-3.738508
11	1.932877	-1.289998	3.179762	-3.807650
12	1.952699	-1.211802	3.131447	-3.859624
13	1.964616	-1.154687	3.096340	-3.897553
14	1.974261	-1.113133	3.070228	-3.925007
15	1.981287	-1.082649	3.051371	-3.945238

La tabla siguiente muestra los resultados de la solución del mismo sistema por el método GS. La solución exacta es $x = (2, -1, 3, -4)$. Se aprecia que en la iteración 15 se tiene una mejor aproximación de la solución con el método de sobrerrelajación.

Gauss-Seidel

k	x_1	x_2	x_3	x_4
0	1.000000	1.000000	1.000000	1.000000
1	5.200000	-3.750000	4.081250	-1.453125
2	2.036250	-3.450781	4.542168	-2.103076
3	1.651746	-3.193777	4.427492	-2.357609
4	1.647204	-2.945539	4.272474	-2.549694
5	1.682025	-2.723966	4.128304	-2.715634
6	1.717631	-2.527427	3.999765	-2.862150
7	1.749749	-2.353270	3.885783	-2.991898
8	1.778274	-2.198968	3.784786	-3.106845
9	1.803554	-2.062259	3.695303	-3.208684
10	1.825953	-1.941139	3.616023	-3.298912
11	1.845798	-1.833828	3.545783	-3.378851
12	1.863381	-1.738753	3.483552	-3.449676
13	1.878958	-1.654519	3.428416	-3.512425
14	1.892760	-1.579890	3.379568	-3.568019
15	1.904987	-1.513770	3.336289	-3.617274

$\diamond$

El método SOR depende de la escogencia de ω y queda entonces la pregunta ¿Cómo escoger ω? La respuesta no es sencilla. Algunas veces se hace simplemente por ensayo y error. Si se desea resolver muchos sistemas de ecuaciones parecidos, por ejemplo provenientes del mismo tipo de problema pero con datos ligeramente diferentes, se puede pensar que un valor adecuado de ω para un problema puede servir para un problema parecido. Entonces se puede pensar en hacer ensayos con varios valores de ω para "ver" y escoger el ω que se supone sirva para este tipo de problemas.

En algunos caso muy particulares se puede hacer un estudio teórico. Tal es el caso de la solución, por diferencias finitas, de la ecuación de Poisson en un rectángulo. Allí se demuestra que

$$\omega_{\text{opt}} = \frac{2}{1 + \sin \dfrac{\pi}{m+1}}$$

Este resultado y otros teóricos se basan en el radio espectral de la matriz de la iteración de punto fijo.

Se puede mostrar que el método SOR se puede expresar como una iteracion

de punto fijo con

$$M_{\text{SOR}} = (D+\omega L)^{-1}\big((1-\omega)D - \omega U\big),$$
$$p_{\text{SOR}} = \omega(D+\omega L)^{-1}b.$$

La deducción anterior proviene de descomponer

$$\begin{aligned} A &= \frac{1}{\omega}D + L \quad + \quad (1-\frac{1}{\omega})D + U \\ &= \frac{1}{\omega}\big(D+\omega L\big) + \frac{1}{\omega}\big((\omega-1)D + \omega U\big) \\ &= \frac{D+\omega L}{\omega} + \frac{(\omega-1)D+\omega U}{\omega} \end{aligned}$$

Entonces

$$\begin{aligned} Ax &= b \\ \big(\frac{D+\omega L}{\omega} + \frac{(\omega-1)D+\omega U}{\omega}\big)x &= b \\ \big(D+\omega L \quad + \quad (\omega-1)D+\omega U\big)x &= \omega b \\ (D+\omega L)x &= -\big((\omega-1)D+\omega U\big)x + \omega b \\ (D+\omega L)x &= \big((1-\omega)D - \omega U\big)x + \omega b \\ x &= (D+\omega L)^{-1}\big((1-\omega)D-\omega U\big)x + \omega(D+\omega L)^{-1}b \end{aligned}$$

Para el caso particular del método GS

$$M_{\text{GS}} = -(D+L)^{-1}U,$$
$$p_{\text{GS}} = (D+L)^{-1}b.$$

Para el ejemplo 3.10, con $\omega = 1.4$,

$$x^{k+1} = \begin{bmatrix} -0.400000 & 0.280000 & -0.560000 & 0.560000 \\ 0.000000 & -0.400000 & -0.700000 & -1.050000 \\ 0.210000 & 0.063000 & 0.261500 & 0.607250 \\ -0.044333 & 0.453367 & 0.583683 & 0.852358 \end{bmatrix} x^k + \begin{bmatrix} 7.000000 \\ -3.500000 \\ 4.287500 \\ -1.799583 \end{bmatrix}.$$

En este caso $\rho(M) = 0.730810$, lo que garantiza la convergencia.

La siguiente tabla nos muestra los valores del número de iteraciones y del radio espectral para diferentes valores de ω. El criterio de parada utilizado fue $\max\{|\delta_i| : i = 1, ..., n\} \leq 0.000001$.

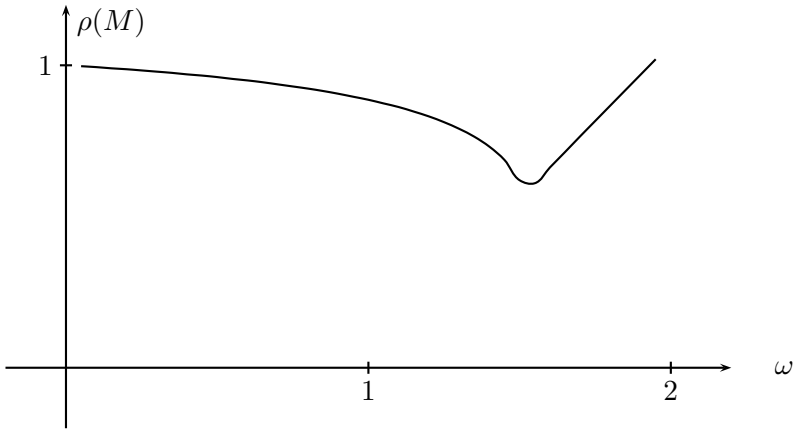


Figura 3.1: Método SOR: ω y radio espectral

ω	k	$\rho(M)$
0.10	999	0.994
0.20	641	0.987
0.30	415	0.979
0.40	301	0.970
0.50	232	0.961
0.60	185	0.950
0.70	151	0.937
0.80	125	0.923
0.90	105	0.906
1.00	88	0.886
1.10	74	0.862
1.20	61	0.831
1.30	50	0.790
1.40	40	0.731
1.50	29	0.620
1.60	33	0.662
1.70	50	0.765
1.80	92	0.867
1.90	408	0.969

La figura 3.1 muestra la variación del radio espectral $\rho(M)$ al variar ω. Proviene de un conjunto de datos más amplio que el de la tabla anterior.

El mejor valor de ω es aproximadamente $\omega \approx 1.55$. Esto coincide, en la tabla, con el menor número de iteraciones.

El siguiente es el esquema del algoritmo de sobrerrelajación, muy parecido al de GS. Se supone que no hay elementos diagonales nulos.

SOR: SOBRERRELAJACIÓN

datos: $A,\ b,\ \omega,\ x^0,\ \varepsilon,$ maxit

$x = x^0$
para $k = 1, ...,$maxit
  difX $= 0$
  para $i = 1, ..., n$
    $r_i = b_i - A_{i\cdot}\, x$
    $\delta_i = \dfrac{r_i}{a_{ii}}$
    $x_i = x_i + \omega\delta_i$
    difX $= \max\{\text{difX},\ |\omega\delta_i|\}$
  fin-para i
  si difX $\leq \varepsilon$ **ent** $x^* \approx x$, **salir**
fin-para k

El método de sobrerrelajación, como el de GS, es útil para sistemas dispersos en los que la matriz se ha almacenado de manera dispersa. Si la matriz es dispersa pero se almacena como si fuera densa, el método de Gauss, en la mayoría de los casos, debe resultar mejor.

3.8 Métodos de minimización

Si A es una matriz simétrica y definida positiva, la solución del sistema

$$Ax = b \tag{3.13}$$

es exactamente el mismo punto x^* que resuelve el siguiente problema de optimización:

$$\min\ f(x) = \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x. \tag{3.14}$$

Como A es definida positiva, entonces f es convexa (más aún, es estrictamente convexa). Para funciones convexas diferenciables, un punto crítico,

punto de gradiente nulo, es necesariamente un minimizador global:

$$\nabla f(x) = f'(x) = Ax - b = 0.$$

Si A es invertible, no necesariamente definida positiva, resolver

$$Ax = b$$

es equivalente a resolver

$$A^{\mathrm{T}}Ax = A^{\mathrm{T}}b$$

y es equivalente a minimizar

$$f(x) = \frac{1}{2}x^{\mathrm{T}}A^{\mathrm{T}}Ax - (A^{\mathrm{T}}b)^{\mathrm{T}}x.$$

La matriz $A^{\mathrm{T}}A$ es definida positiva, luego siempre se puede pensar en resolver un sistema de ecuaciones donde la matriz es definida positiva, problema equivalente a minimizar una función cuadrática estrictamente convexa (3.14).

Para minimizar funciones sin restricciones hay muchos métodos. La mayoría de los método de minimización son iterativos. En casi todos, en cada iteración, dado un punto x^k, hay dos pasos importantes: en el primero se calcula una dirección d^k. Normalmente esta dirección cumple con la propiedad

$$f'(x^k)^{\mathrm{T}}d^k < 0.$$

Esto garantiza que la dirección sea de descenso, es decir, que para t suficientemente pequeño

$$f(x^k + td^k) < f(x^k).$$

El segundo paso consiste en encontrar el mejor t posible, o sea, encontrar

$$t_k = \operatorname{argmin}\ f(x^k + t\,d^k), \quad t \geq 0. \tag{3.15}$$

Con d^k y t_k se construye el siguiente punto

$$x^{k+1} = x^k + t_k\,d^k.$$

Para resolver (3.15) hay varios métodos. Si f es cuadrática (en $\mathbb{R}^n$), entonces $\varphi(t) = f(x^k + td^k)$ es cuadrática (en $\mathbb{R}$). Como A es definida positiva, φ representa una parábola que abre hacia arriba y el punto crítico, t_c, corresponde a un minimizador.

$$\varphi(t) = \frac{1}{2}(x^k + td^k)^{\mathrm{T}} A(x^k + td^k) - b^{\mathrm{T}}(x^k + td^k)$$
$$\varphi(t) = \frac{t^2}{2} d^{k\mathrm{T}} A d^k + t d^{k\mathrm{T}}(Ax^k - b) + f(x^k)$$
$$\varphi'(t) = t d^{k\mathrm{T}} A d^k + d^{k\mathrm{T}}(Ax^k - b)$$

entonces

$$t_k = t_c = -\frac{d^{k\mathrm{T}}(Ax^k - b)}{d^{k\mathrm{T}} A d^k} \tag{3.16}$$

3.9 Método del descenso más pendiente

Un método muy popular, pero no necesariamente muy eficiente, es el método de Cauchy, también llamado método del gradiente o método del descenso más pendiente. En este método la dirección es simplemente el opuesto del gradiente,

$$\begin{aligned} d^k &= -f'(x^k) \\ &= -(Ax^k - b) \end{aligned}$$

Entonces

$$d^k = b - Ax^k \tag{3.17}$$
$$t_k = \frac{d^{k\mathrm{T}} d^k}{d^{k\mathrm{T}} A d^k} \tag{3.18}$$
$$x^{k+1} = x^k + t_k d^k. \tag{3.19}$$

Ejemplo 3.11. Aplicar el método del descenso más pendiente para resolver $Ax = b$, sabiendo que A es definida positiva, donde

$$A = \begin{bmatrix} 4 & 1 & 2 \\ 1 & 5 & -2 \\ 2 & -2 & 10 \end{bmatrix}, \quad b = \begin{bmatrix} 13 \\ -21 \\ 50 \end{bmatrix}, \quad x^0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

```
k = 0
d :     6.000000  -25.000000   40.000000
```

```
t = 0.094488
x1 :     1.566927   -1.362196    4.779514

k = 1
d :    -1.464541   -6.196916   -3.653391
t = 0.190401
x2 :     1.288078   -2.542093    4.083907

k = 2
d :     2.221969   -1.409801    1.500593
t = 0.135469
x3 :     1.589087   -2.733078    4.287191

k = 3
d :     0.802349   -0.349316   -1.516240
t = 0.164510
x4 :     1.721081   -2.790544    4.037754
k = 4
d :     0.830711   -0.692854    0.599209
t = 0.135907
x5 :     1.833980   -2.884707    4.119191

k = 5
d :     0.310405   -0.172063   -0.629281
t = 0.164543
x6 :     1.885055   -2.913019    4.015647

x7  :     1.931468   -2.952251    4.049268
x8  :     1.952504   -2.964045    4.006467
x9  :     1.971680   -2.980265    4.020361
x10 :     1.980371   -2.985141    4.002673
x11 :     1.988296   -2.991844    4.008415
x12 :     1.991888   -2.993859    4.001105
x13 :     1.995163   -2.996629    4.003477
x14 :     1.996648   -2.997462    4.000456
x15 :     1.998001   -2.998607    4.001437
x16 :     1.998615   -2.998951    4.000189
x17 :     1.999174   -2.999424    4.000594
x18 :     1.999427   -2.999567    4.000078
x19 :     1.999659   -2.999762    4.000245
```

```
x20 :     1.999763   -2.999821    4.000032
```

Ejemplo 3.12. Aplicar el método del descenso más pendiente para resolver $Ax = b$, sabiendo que A es definida positiva, donde

$$A = \begin{bmatrix} 19 & 6 & 8 \\ 6 & 5 & 2 \\ 8 & 2 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 55 \\ 22 \\ 24 \end{bmatrix}, \quad x^0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

```
k = 0
d :    22.000000    9.000000   10.000000
t = 0.040905
x1 :     1.899920    1.368149    1.409055

k = 1
d :    -0.579812    0.941625    0.428123
t = 0.531990
x2 :     1.591466    1.869085    1.636812

k = 2
d :     0.453147   -0.167842    0.982857
t = 0.089118
x3 :     1.631849    1.854127    1.724402

k = 3
d :    -0.925117   -0.510535    0.339342
t = 0.068514
x4 :     1.568466    1.819148    1.747652

k = 4
d :     0.303036   -0.001843    0.823366
t = 0.091249
x5 :     1.596118    1.818980    1.822783

k = 5
d :    -0.822384   -0.317174    0.301965
t = 0.069496
x6 :     1.538966    1.796938    1.843768

x95  :     1.025125    1.989683    2.952309
```

```
x96  :     1.022738    1.989417    2.953040
x97  :     1.023406    1.990389    2.955571
x98  :     1.021183    1.990141    2.956253
x99  :     1.021805    1.991047    2.958611
x100 :     1.019734    1.990816    2.959245
x101 :     1.020313    1.991659    2.961442
```

La rapidez de convergencia del método del descenso más pendiente, cuando A es definida positiva, depende del cociente $\dfrac{\lambda_n}{\lambda_1}$, donde λ_n es el mayor valor propio y λ_1 el menor. Si el cociente es cercano a uno, hay buena convergencia. Si el cociente es grande la convergencia es lenta a causa del zigzagueo.

```
primer ejemplo

 v  =
    2.3714059
    5.5646277
    11.063966

 coc  =    4.6655726

Segundo ejemplo:
 valores propios
    0.4250900
    3.0722446
    24.502665
 coc  =    57.641129
```

3.10 Método del gradiente conjugado

Dentro del grupo de métodos de direcciones conjugadas, está el método del gradiente conjugado. Este método se adapta muy bien cuando la matriz es "dispersa". Tiene la ventaja adicional que, aunque es un método iterativo, a lo más en n iteraciones se obtiene la solución exacta, si no hay rerrores de redondeo.

En el método GC la dirección se construye agregando a $-f'(x^k)$ un múltiplo

de la dirección anterior,

$$d^k = -f'(x^k) + \alpha_k \, d^{k-1}. \tag{3.20}$$

Dos direcciones diferentes, d^i y d^j, se llaman conjugadas con respecto a A si

$$d^{i^{\mathrm{T}}} A \, d^j = 0.$$

Para el caso de la solución de un sistema lineal por medio del método GC, es corriente denominar el vector residuo

$$r^k = Ax^k - b. \tag{3.21}$$

Obviamente $x^k = x^*$ si y solamente si $r^k = 0$. El vector residuo es exactamente el mismo gradiente de f en el punto x^k.

Las fórmulas que completan la definición del método GC son:

$$\alpha_1 = 0, \tag{3.22}$$

$$\alpha_k = \frac{||r^k||_2^2}{||r^{k-1}||_2^2}, \quad k = 2, ..., n, \tag{3.23}$$

$$t_k = \frac{||r^k||_2^2}{d^{k^{\mathrm{T}}} A d^k}, \quad k = 1, ..., n. \tag{3.24}$$

Suponiendo que A es definida positiva, el método GC tiene las siguientes propiedades:

- d^k es dirección de descenso.
- $f(x^k) < f(x^{k-1})$.
- las direcciones son conjugadas con respecto a A.
- Si no hay errores de redondeo, entonces $x^* = x^k$ para algún $k \leq n+1$.

Cuando se llega a x^{n+1} y no se obtiene la solución con la precisión deseada, entonces se vuelve a empezar el proceso utilizando como nuevo x^1 el x^{n+1} obtenido.

MÉTODO DEL GRADIENTE CONJUGADO

datos: A, b, x^1, $\texttt{MAXIT}$, ε
para $K = 1, ..., \texttt{MAXIT}$
  para $k = 1, ..., n$
    $r^k = Ax^k - b$
    si $||r^k|| < \varepsilon$ **ent parar**
    si $k = 1$ **ent** $d^k = -r^k$
    sino
      $\alpha_k = \dfrac{||r^k||_2^2}{||r^{k-1}||_2^2}$
      $d^k = -r^k + \alpha_k d^{k-1}$
    fin-sino
    $t_k = \dfrac{||r^k||_2^2}{{d^k}^{\mathrm{T}} A d^k}$
    $x^{k+1} = x^k + t_k d^k$
  fin-para k
  $x^1 = x^{n+1}$
fin-para K

Ejemplo 3.13. Resolver el sistema $Ax = b$ por el método GC, partiendo de $x^1 = (1, 1, 1)$, donde

$$A = \begin{bmatrix} 19 & 6 & 8 \\ 6 & 5 & 2 \\ 8 & 2 & 4 \end{bmatrix}, \qquad b = \begin{bmatrix} 55 \\ 22 \\ 24 \end{bmatrix}.$$

$$
\begin{aligned}
r^1 &= Ax^1 - b = (-22, -9, -10),\\
||r^1||_2^2 &= 665,\\
d^1 &= -r^1 = (22, 9, 10),\\
{d^1}^{\mathrm{T}} A d^1 &= 16257,\\
t_1 &= \frac{665}{16257} = 0.040905,\\
x^2 &= x^1 + t_1 d^1 = (1.899920, 1.368149, 1.409055),
\end{aligned}
$$

$$
\begin{aligned}
r^2 &= (0.579812, -0.941625, -0.428123),\\
||r^2||_2^2 &= 1.406129,\\
\alpha_2 &= \frac{1.406129}{665} = 0.002114,\\
d^2 &= (-0.533293, 0.960655, 0.449268),\\
{d^2}^{\mathrm{T}} A d^2 &= 2.570462,\\
t_2 &= 0.547034,\\
x^3 &= (1.608191, 1.893660, 1.654819),
\end{aligned}
$$

$$
\begin{aligned}
r^3 &= (0.156138, 0.427083, -0.727877),\\
||r^3||_2^2 &= 0.736584,\\
\alpha_3 &= 0.523838,\\
d^3 &= (-0.435497, 0.076145, 0.963221),\\
{d^3}^{\mathrm{T}} A d^3 &= 0.527433,\\
t_3 &= 1.396545,\\
x^4 &= (1, 2, 3),
\end{aligned}
$$

$$
\begin{aligned}
x^1 &= x^4 = (1, 2, 3),\\
r^1 &= (0, 0, 0).
\end{aligned}
$$

Si la matriz A es dispersa y se utiliza una estructura de datos donde solamente se almacenen los elementos no nulos, para poder implementar con éxito el método GC, se requiere simplemente poder efectuar el producto de la matriz A por un vector. Hay dos casos, Ax^k para calcular r^k y Ad^k

para calcular t_k. Las otras operaciones necesarias son producto escalar entre vectores, sumas o restas de vectores y multiplicación de un escalar por un vector. Todo esto hace que sea un método muy útil para matrices muy grandes pero muy poco densas.

4

Solución de ecuaciones no lineales

Uno de los problemas más corrientes en matemáticas consiste en resolver una ecuación, es decir, encontrar un valor $x^* \in \mathbb{R}$ que satisfaga

$$f(x) = 0,$$

donde f es una función de variable y valor real, o sea,

$$f : \mathbb{R} \to \mathbb{R}.$$

Este x^* se llama solución de la ecuación. A veces también se dice que x^* es una raíz o un cero. Algunos ejemplos sencillos de ecuaciones son:

$$\begin{aligned} x^5 - 3x^4 + 10x - 8 &= 0, \\ e^x - x^3 + 8 &= 0, \\ \frac{x^2 + x}{\cos(x-1) + 2} - x &= 0. \end{aligned}$$

En algunos casos no se tiene una expresión sencilla de f, sino que $f(x)$ corresponde al resultado de un proceso; por ejemplo:

$$\int_{-\infty}^{x} e^{-t^2} dt - 0.2 = 0.$$

Lo mínimo que se le exige a f es que sea continua. Si no es continua en todo $\mathbb{R}$, por lo menos debe ser continua en un intervalo $[a, b]$ donde se busca

la raíz. Algunos métodos requieren que f sea derivable. Para la aplicación de algunos teoremas de convergencia, no para el método en sí, se requieren derivadas de orden superior.

Los métodos generales de solución de ecuaciones sirven únicamente para hallar raíces reales. Algunos métodos específicos para polinomios permiten obtener raíces complejas.

Los métodos presuponen que la ecuación $f(x) = 0$ tiene solución. Es necesario, antes de aplicar mecánicamente los métodos, estudiar la función, averiguar si tiene raíces, ubicarlas aproximadamente. En algunos casos muy difíciles no es posible hacer un análisis previo de la función, entonces hay que utilizar de manera mecánica uno o varios métodos, pero sabiendo que podrían ser ineficientes o, simplemente, no funcionar.

La mayoría de los métodos parten de x_0, aproximación inicial de x^*, a partir del cual se obtiene x_1. A partir de x_1 se obtiene x_2, después x_3, y así sucesivamente se construye la sucesión $\{x_k\}$ con el objetivo, no siempre cumplido, de que

$$\lim_{k\to\infty} x_k = x^*.$$

El proceso anterior es teóricamente infinito, y obtendría la solución después de haber hecho un número infinito de cálculos. En la práctica el proceso se detiene cuando se obtenga una aproximación suficientemente buena de x^*. Esto querría decir que el proceso se detendría cuando

$$|x_k - x^*| \leq \varepsilon,$$

para un ε dado. El anterior criterio supone el conocimiento de x^*, que es justamente lo buscado. Entonces se utiliza el criterio, éste si aplicable,

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

En la mayoría de los casos, cuanto más cerca esté x_0 de x^*, más rápidamente se obtendrá una buena aproximación de x^*.

Otros métodos parten de un intervalo inicial $[a_0, b_0]$, en el cual se sabe que existe una raíz x^*. A partir de él, se construye otro intervalo $[a_1, b_1]$, contenido en el anterior, en el que también está x^* y que es de menor tamaño. De manera análoga se construye $[a_2, b_2]$. Se espera que la sucesión formada

por los tamaños tienda a 0. Explícitamente,

$$\begin{aligned} x^* &\in [a_0, b_0], \\ [a_{k+1}, b_{k+1}] &\subset [a_k, b_k],\ k = 1, 2, ..., \\ x^* &\in [a_k, b_k],\ k = 1, 2, ..., \\ \lim_{k\to\infty} (b_k - a_k) &= 0. \end{aligned}$$

En este caso, el proceso se detiene cuando se obtiene un intervalo suficientemente pequeño,

$$|b_k - a_k| \leq \varepsilon.$$

Cualquiera de los puntos del último intervalo es una buena aproximación de x^*.

4.1 En Scilab

Para resolver

$$f(x) = 0,$$

donde f es una función de variable y valor real, se utiliza `fsolve`. Por ejemplo para resolver

$$\frac{x - e^x}{1 + x^2} - \cos(x) + 0.1 = 0,$$

es necesario definir una función de Scilab donde esté f y después utilizar `fsolve`.

```
function fx = func156(x)
  fx = ( x - exp(x) )/( 1 + x*x ) - cos(x) + 0.1
endfunction
```

Despues de haber cargado esta función, se utiliza `fsolve` dándole como parámetros, la aproximación inicial y la función:

```
r = fsolve(0, func156)
```

Con otra aproximación inicial podría dar otra raíz. Un parámetro opcional, que puede acelerar la obtención de la solución, es otra función de Scilab donde esté definida la derivada.

```
function y = f123(x)
  y = x*x*x - 4*x*x + 10*x - 20
endfunction
//-----------------------------------------------
function d1 = der123(x)
  d1 = 3*x*x - 8*x +10
endfunction
```

La orden de Scilab puede ser semejante a `fsolve(1, f123, der123)`. Claramente es más cómodo no tener que definir la derivada, pero no hacerlo puede hacer menos eficiente el uso de `fsolve`.

La función `fsolve` trabaja bien pero no siempre encuentra una solución. Por ejemplo,

```
function y = f13(x)
  y = exp(x) - 2.7*x
endfunction

x  = fsolve(1, f13)
```

da como resultado `0.9933076`. Lo anterior hará que el usuario ingenuamente suponga que ese valor corresponde a una raíz. Realmente la función no tiene raíces. Es conveniente utilizar `fsolve` con tres parámetros de salida,

```
[x, fx, info] = fsolve(1, f13)
```

`fx` será el valor de `f13` evaluada en `x`, e `info` valdrá 1 si se obtuvo la solución con la precisión deseada. Para nuestro ejemplo los valores serán

```
info  =

   4.
fx  =

   0.0182285
x  =

   0.9957334
```

lo cual indica que no se obtuvo una raíz.

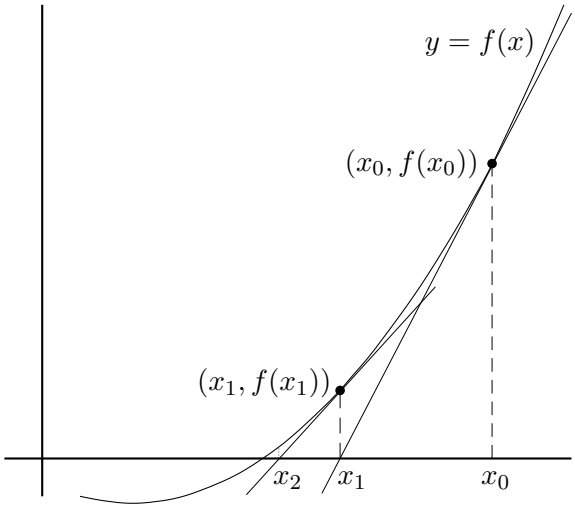


Figura 4.1: Método de Newton

4.2 Método de Newton

También se conoce como el método de Newton-Raphson. Dado x_0, se construye la recta tangente en $(x_0, f(x_0))$. El valor de x donde esta recta corta el eje x es el nuevo valor x_1. Ahora se construye la recta tangente en el punto $(x_1, f(x_1))$. El punto de corte entre la recta y el eje x determina x_2...

En el caso general, dado x_k, se construye la recta tangente en el punto $(x_k, f(x_k))$,

$$y = f'(x_k)(x - x_k) + f(x_k).$$

Para $y = 0$ se tiene $x = x_{k+1}$,

$$0 = f'(x_k)(x_{k+1} - x_k) + f(x_k).$$

Entonces

$$\boxed{x_{k+1} \;=\; x_k - \frac{f(x_k)}{f'(x_k)}} \tag{4.1}$$

Ejemplo 4.1. Aplicar el método de Newton a la ecuación $x^5 - 3x^4 + 10x - 8 = 0$, partiendo de $x_0 = 3$.

k	x_k	$f(x_k)$	$f'(x_k)$
0	3.000000	2.200000E+01	91.000000
1	2.758242	5.589425E+00	47.587479
2	2.640786	9.381331E-01	32.171792
3	2.611626	4.892142E-02	28.848275
4	2.609930	1.590178E-04	28.660840
5	2.609924	1.698318E-09	28.660228
6	2.609924	-2.838008E-15	28.660227

Las raíces reales del polinomio $x^5 - 3x^4 + 10x - 8$ son: 2.6099, 1.3566, 1. Tomando otros valores iniciales el método converge a estas raíces. Si se toma $x_0 = 2.1$, se esperaría que el método vaya hacia una de las raíces cercanas, 2.6099 o 1.3566 . Sin embargo, hay convergencia hacia 1.

k	x_k	$f(x_k)$	$f'(x_k)$
0	2.100000	-4.503290e+00	-3.891500
1	0.942788	-1.974259e-01	3.894306
2	0.993484	-1.988663e-02	3.103997
3	0.999891	-3.272854e-04	3.001745
4	1.000000	-9.509814e-08	3.000001
5	1.000000	-7.993606e-15	3.000000

El método de Newton es muy popular por sus ventajas:

- Sencillez.
- Generalmente converge.
- En la mayoría de los casos, cuando converge, lo hace rápidamente.

También tiene algunas desventajas:

- Puede no converger.
- Presenta problemas cuando $f'(x_k) \approx 0$.
- Requiere poder evaluar, en cada iteración, el valor $f'(x)$.

La implementación del método de Newton debe tener en cuenta varios aspectos. Como no es un método totalmente seguro, debe estar previsto un

número máximo de iteraciones, llamado por ejemplo `maxit`. Para una precisión ε_f, la detención deseada para el proceso iterativo se tiene cuando $|f(x_k)| \leq \varepsilon_f$. Otra detención posible se da cuando dos valores de x son casi iguales, es decir, cuando $|x_k - x_{k-1}| \leq \varepsilon_x$. Se acostumbra a utilizar el cambio relativo, o sea, $|x_k - x_{k-1}|/|x_k| \leq \varepsilon_x$. Para evitar las divisiones por cero, se usa $|x_k - x_{k-1}|/(1 + |x_k|) \leq \varepsilon_x$. Finalmente, siempre hay que evitar las divisiones por cero o por valores casi nulos. Entonces, otra posible parada, no deseada, corresponde a $|f'(x_k)| \leq \varepsilon_0$. El algoritmo para el método de Newton puede tener el siguiente esquema:

datos: `x0`, `maxit`, ε_f, ε_x, ε_0
`xk = x0`
`fx =` f`(xk)`, `fpx =` f'`(xk)`
para `k=1,...,maxit`
    si `|fpx|` $\leq \varepsilon_0$ **ent salir**
    δ `= fx/fpx`
    `xk = xk-`δ
    `fx =` f`(xk)`, `fpx =` f'`(xk)`
    si `|fx|` $\leq \varepsilon_f$ **ent salir**
    si `|`δ`|/(1+|xk|)` $\leq \varepsilon_x$ **ent salir**
fin-para `k`

Para la implementación en Scilab, es necesario determinar cómo se evalúa f y f'. Fundamentalmente hay dos posibilidades:

- Hacer una función para evaluar f y otra para evaluar f'.
- Hacer una función donde se evalúe al mismo tiempo f y f'.

En la siguiente implementación del método de Newton, la función `f` debe evaluar al mismo tiempo $f(x)$ y $f'(x)$.

```
function [fx, dfx] = f321(x)
   fx  = x^5 - 3*x^4 + 10*x - 8
   dfx = 5*x^4 -12*x^3 + 10
endfunction
//-------------------------------------------------------------
function [x, ind] = Newton(func, x0,  eps, maxit)
  // metodo de Newton
```

```
// func  debe dar los valores f(x)  y  f'(x)

// ind valdra   1  si se obtiene la raiz
//              2  si se hicieron muchas iteraciones, > maxit
//              0  si una derivada es nula o casi
//
//*************
eps0 = 1.0e-12
//*************

x = x0
for k=0:maxit
  [fx, der] = func(x)
  //printf('%3d  %10.6f %10.6f %10.6f\n', k, x, fx, der)

  if abs(fx) <= eps
    ind = 1
    return
  end

  if abs(der) <= eps0
    ind = 0
    return
  end

  x = x - fx/der
end
ind = 2
endfunction
```

El llamado puede ser semejante a

```
[x, result] = Newton(f321, 3, 1.0e-8, 20)
```

4.2.1 Orden de convergencia

Teorema 4.1. *Sean $a < b$, $I =]a, b[$, $f : I \to \mathbb{R}$, $x^* \in I$, $f(x^*) = 0$, f' y f'' existen y son continuas en I, $f'(x^*) \neq 0$, $\{x_k\}$ la sucesión definida por 4.1.*

Si x_0 está suficientemente cerca de x^*, entonces

$$\lim_{k\to\infty} x_k = x^*, \tag{4.2}$$

$$\lim_{k\to\infty} \frac{|x_{k+1}-x^*|}{|x_k-x^*|^2} = \frac{|f''(x^*)|}{2|f'(x^*)|} \tag{4.3}$$

El primer resultado dice que la sucesión converge a x^*. El segundo dice que la convergencia es cuadrática o de orden superior. La frase "x_0 está suficientemente cerca de x^*, entonces..." quiere decir que existe $\varepsilon > 0$ tal que si $x_0 \in [x^*-\varepsilon, x^*+\varepsilon] \subseteq I$, entonces...

Demostración.

$$f(x) = f(x_n) + f'(x_n)(x-x_n) + f''(\xi)\frac{(x-x_n)^2}{2}\ , \quad \xi \in I(x, x_n)$$

tomando $x = x^*$

$$f(x^*) = 0 = f(x_n) + f'(x_n)(x^*-x_n) + f''(\xi)\frac{(x^*-x_n)^2}{2}\ , \quad \xi \in I(x^*, x_n)$$

dividiendo por $f'(x_n)$

$$\begin{aligned}
0 &= \frac{f(x_{xn})}{f'(x_{xn})} + (x^*-x_n) + (x^*-x_n)^2\frac{f''(\xi)}{2f'(x_n)}\ , \\
0 &= x^* - \left(x_n - \frac{f(x_{xn})}{f'(x_{xn})}\right) + (x^*-x_n)^2\frac{f''(\xi)}{2f'(x_n)}\ , \\
0 &= x^* - x_{n+1} + (x^*-x_n)^2\frac{f''(\xi)}{2f'(x_n)}\ , \\
x^* - x_{n+1} &= -(x^*-x_n)^2\frac{f''(\xi)}{2f'(x_n)}\ .
\end{aligned} \tag{4.4}$$

Sea

$$I = [x^*-\varepsilon, x+\varepsilon]$$

$$M = \frac{\max\limits_{x\in I}|f''(x)|}{2\min\limits_{x\in I}|f'(x)|}$$

Como $f'(x^*) \neq 0$ y f' es continua, se puede escoger ε suficientemente pequeño para que $\min_{x\in I} |f'(x)| > 0$. A partir de (4.4) se obtiene

$$|x^* - x_{n+1}| \leq M|x^* - x_n|^2 \,. \tag{4.5}$$

En particular,

$$\begin{aligned} |x^* - x_1| &= M|x^* - x_0|^2 \,, \\ M|x^* - x_1| &= (M|x^* - x_0|)^2 \,. \end{aligned}$$

Sea x_0 tal que

$$\begin{aligned} |x^* - x_0| &< \varepsilon, \\ M|x^* - x_0| &< 1 \,. \end{aligned}$$

Entonces

$$\begin{aligned} &M|x^* - x_1| < 1 \,, \\ &M|x^* - x_1| < (M|x^* - x_0|)^2, \\ &M|x^* - x_1| < M|x^* - x_0|, \quad \text{ya que } 0 < t < 1 \;\Rightarrow\; t^2 < t, \\ &\quad |x^* - x_1| < \varepsilon, \\ &\qquad \vdots \\ &M|x^* - x_n| < 1 \,, \\ &\quad |x^* - x_n| < \varepsilon \,. \end{aligned}$$

Luego

$$\begin{aligned} |x^* - x_n| &\leq M|x^* - x_{n-1}|, \\ M|x^* - x_n| &\leq (M|x^* - x_{n-1}|)^2, \\ M|x^* - x_n| &\leq (M|x^* - x_0|)^{2^n}, \\ |x^* - x_n| &\leq \frac{1}{M}(M|x^* - x_0|)^{2^n}, \end{aligned}$$

Como $|x^* - x_0| < 1$, entonces

$$\lim_{n\to\infty} |x^* - x_n| = 0,$$

es decir

$$\lim_{n\to\infty} x_n = x^* .$$

Reescribiendo (4.4),

$$\frac{x^* - x_{n+1}}{(x^* - x_n)^2} = -\frac{f''(\xi)}{2f'(x_n)}, \quad \xi \in I(x^*, x_n)$$

Tomando el límite, como x_n tiende a x^*,

$$\lim_{n\to\infty} \frac{x^* - x_{n+1}}{(x^* - x_n)^2} = -\frac{f''(x^*)}{2f'(x^*)} .$$

A manera de comprobación, después de que se calculó una raíz, se puede ver si la sucesión muestra aproximadamente convergencia cuadrática. Sea $e_k = x_k - x^*$. La sucesión $|e_k|/|e_{k-1}|^2$ debería acercarse a $|f''(x^*)|/\ (2|f'(x^*)|)$. Para el ejemplo anterior $|f''(x^*)/(2|f'(x^*)|) = 16/(2 \times 3) = 2.6666$.

k	x_k	$\lvert e_k\rvert$	$\lvert e_k\rvert/\lvert e_{k-1}\rvert^2$
0	2.1000000000000001	1.100000e+00	
1	0.9427881279712185	5.721187e-02	4.728254e-02
2	0.9934841559110774	6.515844e-03	1.990666e+00
3	0.9998909365826297	1.090634e-04	2.568844e+00
4	0.9999999683006239	3.169938e-08	2.664971e+00
5	0.9999999999999973	2.664535e-15	2.651673e+00

4.3 Método de la secante

Uno de los inconvenientes del método de Newton es que necesita evaluar $f'(x)$ en cada iteración. Algunas veces esto es imposible o muy difícil. Si en el método de Newton se modifica la fórmula 4.1 reemplazando $f'(x_k)$ por una aproximación

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

entonces se obtiene

$$\boxed{x_{k+1} \ = \ x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}} \tag{4.6}$$

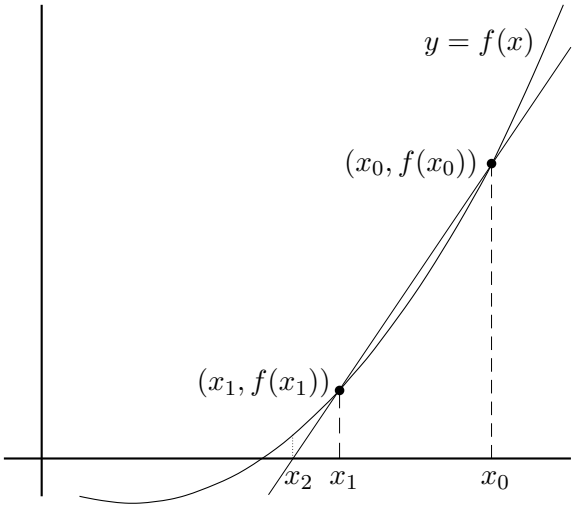


Figura 4.2: Método de la secante

En el método de Newton se utilizaba la recta tangente a la curva en el punto $(x_k, f(x_k))$. En el método de la secante se utiliza la recta (secante) que pasa por los puntos $(x_k, f(x_k))$ y $(x_{k-1}, f(x_{k-1}))$.

Ejemplo 4.2. Aplicar el método de la secante a la ecuación $x^5 - 3x^4 + 10x - 8 = 0$, partiendo de $x_0 = 3$.

k	x_k	$f(x_k)$
0	3.000000	2.200000e+01
1	3.010000	2.292085e+01
2	2.761091	5.725624e+00
3	2.678210	2.226281e+00
4	2.625482	4.593602e-01
5	2.611773	5.317368e-02
6	2.609979	1.552812e-03
7	2.609925	5.512240e-06
8	2.609924	5.747927e-10
9	2.609924	-2.838008e-15

Mediante condiciones semejantes a las exigidas en el teorema 4.1 se muestra (ver [Sch91]), que el método de la secante tiene orden de convergencia

$$\frac{1+\sqrt{5}}{2} \approx 1.618$$

Como el método de la secante es semejante al método de Newton, entonces tienen aproximadamente las mismas ventajas y las mismas desventajas, salvo dos aspectos:

- La convergencia del método de la secante, en la mayoría de los casos, es menos rápida que en el método de Newton.
- El método de la secante obvia la necesidad de evaluar las derivadas.

El esquema del algoritmo es semejante al del método de Newton. Hay varias posibles salidas, algunas deseables, otras no.

datos: x0, maxit, ε_f, ε_x, ε_0
x1 = x0 + 0.1, f0 = f(x0), f1 = f(x1)
para k=1,...,maxit
    den = f1-f0
    si |den| $\leq \varepsilon_0$ **ent salir**
    δ =f1*(x1-x0)/den
    x2 = x1 - δ, f2 = f(x2)
    si |f2| $\leq \varepsilon_f$ **ent salir**
    si |δ|/(1+|x2|) $\leq \varepsilon_x$ **ent salir**
    x0 = x1, f0 = f1, x1 = x2, f1 = f2
fin-para k

El método de la secante se puede implementar en Scilab así:

```
function [x, ind] = secante(f, x0, epsx, epsf, maxit)
  // metodo de la secante

  // ind valdra   1  si se obtiene la raiz,
  //                 | f(x2) | < epsf  o
  //                 | x2-x1 | < epsx
  //
  //              2  si se hicieron muchas iteraciones, > maxit
  //              0  si un denominador es nulo o casi nulo

  //*************
  eps0 = 1.0e-12
  //*************
```

```
  x = x0

  h  = 0.1
  x1 = x0 + h
  f0 = f(x0)
  f1 = f(x1)

  for k=1:maxit
    den = f1-f0
    if abs(den) <= eps0
      ind = 0
      return
    end
    d2 = f1*(x1-x0)/den
    x2 = x1 - d2
    f2 = f(x2)
    disp(k,x2,f2)
    if abs(f2) <= epsf | abs(d2) <= epsx
      x = x2
      ind = 1
      return
    end
    x0 = x1, f0 = f1
    x1 = x2, f1 = f2
  end
  x = x2
  ind = 2
endfunction
```

4.4 Método de la bisección

Si la función f es continua en el intervalo $[a, b]$, $a < b$, y si $f(a)$ y $f(b)$ tienen signo diferente,

$$f(a)f(b) < 0,$$

entonces f tiene por lo menos una raíz en el intervalo. Este método ya se vio en el capítulo sobre funciones.

Usualmente se define el error asociado a una aproximación como

$$e_k = |x_k - x^*|.$$

En el método de la bisección, dado el intervalo $[a_k, b_k]$, $a_k < b_k$, no se tiene un valor de x_k. Se sabe que en $[a_k, b_k]$ hay por lo menos una raíz. Cualquiera de los valores en el intervalo podría ser x_k. Sea E_k el máximo error que puede haber en la iteración k,

$$e_k \leq E_k = b_k - a_k.$$

Como el tamaño de un intervalo es exactamente la mitad del anterior

$$b_k - a_k = \frac{1}{2}(b_{k-1} - a_{k-1}),$$

entonces

$$b_k - a_k = \left(\frac{1}{2}\right)\left(\frac{1}{2}\right)(b_{k-2} - a_{k-2}).$$

Finalmente

$$b_k - a_k = \left(\frac{1}{2}\right)^k (b_0 - a_0).$$

Obviamente $E_k \to 0$ y

$$\frac{E_k}{E_{k-1}} = \frac{1}{2} \to \frac{1}{2}.$$

Esto quiere decir que la sucesión de cotas del error tiene convergencia lineal (orden 1) y tasa de convergencia 1/2.

En el método de la bisección se puede saber por anticipado el número de iteraciones necesarias para obtener un tamaño deseado,

$$\begin{aligned}
b_k - a_k &\leq \varepsilon, \\
\left(\frac{1}{2}\right)^k (b_0 - a_0) &\leq \varepsilon, \\
\left(\frac{1}{2}\right)^k &\leq \frac{\varepsilon}{b_0 - a_0}, \\
2^k &\geq \frac{b_0 - a_0}{\varepsilon}, \\
k \log 2 &\geq \log \frac{b_0 - a_0}{\varepsilon}, \\
k &\geq \frac{\log \frac{b_0 - a_0}{\varepsilon}}{\log 2}.
\end{aligned}$$

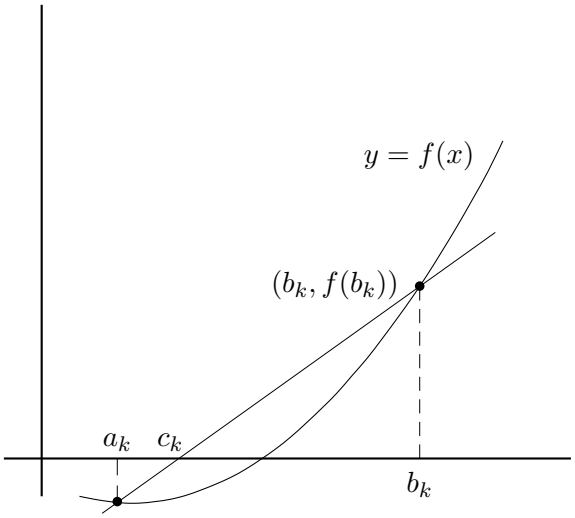


Figura 4.3: Método *Regula Falsi*

Por ejemplo, si el tamaño del intervalo inicial es 3, si $\varepsilon = 1.0E-6$, entonces en $k = 22$ (≥ 21.52) iteraciones se obtiene un intervalo suficientemente pequeño.

4.5 Método de Regula Falsi

Igualmente se conoce con el nombre de falsa posición. Es una modificación del método de la bisección. También empieza con un intervalo $[a_0, b_0]$ donde f es continua y tal que $f(a_0)$ y $f(b_0)$ tienen signo diferente.

En el método de bisección, en cada iteración, únicamente se tiene en cuenta el signo de $f(a_k)$ y de $f(b_k)$, pero no sus valores: no se está utilizando toda la información disponible. Además es de esperar que si $f(a_k)$ está más cerca de 0 que $f(b_k)$, entonces puede ser interesante considerar, no el punto medio, sino un punto más cercano a a_k. De manera análoga, si $f(b_k)$ está más cerca de 0 que $f(a_k)$, entonces puede ser interesante considerar, no el punto medio, sino un punto más cercano a b_k.

En el método de Regula Falsi se considera el punto donde la recta que pasa por $(a_k, f(a_k))$, $(b_k, f(b_k))$ corta el eje x. Como $f(a_k)$ y $f(b_k)$ tienen signo diferente, entonces el punto de corte c_k queda entre a_k y b_k.

La ecuación de la recta es:

$$y - f(a_k) = \frac{f(b_k) - f(a_k)}{b_k - a_k}(x - a_k)$$

Cuando $y = 0$ se tiene el punto de corte $x = c_k$,

$$\boxed{c_k = a_k - \frac{f(a_k)(b_k - a_k)}{f(b_k) - f(a_k)}} \tag{4.7}$$

Esta fórmula es semejante a la de la secante. Como $f(a_k)$ y $f(b_k)$ tienen signo diferente, entonces $f(b_k) - f(a_k)$ tiene signo contrario al de $f(a_k)$. Entonces $-f(a_k)/(f(b_k) - f(a_k)) > 0$. Usando de nuevo que $f(a_k)$ y $f(b_k)$ tienen signo diferente, entonces $|f(a_k)|/|f(b_k) - f(a_k)| < 1$. Luego $0 < -f(a_k)/(f(b_k) - f(a_k)) < 1$. Esto muestra que $a_k < c_k < b_k$.

Partiendo de un intervalo inicial $[a_0, b_0]$, en la iteración k se tiene el intervalo $[a_k, b_k]$ donde f es continua y $f(a_k)$, $f(b_k)$ tienen diferente signo. Se calcula c_k el punto de corte y se tienen tres posibilidades excluyentes:

- $f(c_k) \approx 0$; en este caso c_k es, aproximadamente, una raíz;
- $f(a_k)f(c_k) < 0$; en este caso hay una raíz en el intervalo $[a_k, c_k] = [a_{k+1}, b_{k+1}]$;
- $f(a_k)f(c_k) > 0$; en este caso hay una raíz en el intervalo $[c_k, b_k] = [a_{k+1}, b_{k+1}]$.

Ejemplo 4.3. Aplicar el método de Regula Falsi a la ecuación $x^5 - 3x^4 + 10x - 8 = 0$, partiendo de $[2, 5]$.

k	a_k	b_k	$f(a_k)$	$f(b_k)$	c_k	$f(c_k)$
0	2.000000	5	-4.000000	1292	2.009259	-4.054857
1	2.009259	5	-4.054857	1292	2.018616	-4.108820
2	2.018616	5	-4.108820	1292	2.028067	-4.161744
3	2.028067	5	-4.161744	1292	2.037610	-4.213478
4	2.037610	5	-4.213478	1292	2.047239	-4.263862
5	2.047239	5	-4.263862	1292	2.056952	-4.312734
10	2.096539	5	-4.489666	1292	2.106594	-4.528370
20	2.198548	5	-4.739498	1292	2.208787	-4.744664
30	2.298673	5	-4.594020	1292	2.308244	-4.554769
335	2.609924	5	-0.000001	1292	2.609924	-0.000001

Como se ve, la convergencia es muy lenta. El problema radica en que en el método de Regula Falsi **no se puede garantizar** que

$$\lim_{k\to\infty}(b_k - a_k) = 0.$$

Esto quiere decir que el método no es seguro. Entonces, en una implementación, es necesario trabajar con un número máximo de iteraciones.

4.6 Modificación del método de Regula Falsi

Los dos métodos, bisección y Regula Falsi, se pueden combinar en uno solo de la siguiente manera. En cada iteración se calcula m_k y c_k. Esto define tres subintervalos en $[a_k, b_k]$. En por lo menos uno de ellos se tiene una raíz. Si los tres subintervalos sirven, se puede escoger cualquiera, o mejor aún, el de menor tamaño. En un caso muy especial, cuando m_k y c_k coinciden, se tiene simplemente una iteración del método de bisección.

En cualquiera de los casos

$$b_{k+1} - a_{k+1} \leq \frac{1}{2}(b_k - a_k),$$

entonces

$$b_k - a_k \leq \left(\frac{1}{2}\right)^k (b_0 - a_0),$$

lo que garantiza que

$$\lim_{k\to\infty}(b_k - a_k) = 0.$$

Ejemplo 4.4. Aplicar la modificación del método de Regula Falsi a la ecuación $x^5 - 3x^4 + 10x - 8 = 0$, partiendo de $[2, 5]$.

k	a	b	f(a)	f(b)	c	f(c)	m	f(m)
0	2.0000	5.0000	-4.00e+0	1.29e+3	2.0093	-4.0e+0	3.5000	1.0e+2
1	2.0093	3.5000	-4.05e+0	1.02e+2	2.0662	-4.4e+0	2.7546	5.4e+0
2	2.0662	2.7546	-4.36e+0	5.42e+0	2.3731	-4.2e+0	2.4104	-3.8e+0
3	2.4104	2.7546	-3.80e+0	5.42e+0	2.5523	-1.5e+0	2.5825	-7.4e-1
4	2.5825	2.7546	-7.44e-1	5.42e+0	2.6033	-1.9e-1	2.6686	1.9e+0
5	2.6033	2.6686	-1.87e-1	1.88e+0	2.6092	-2.0e-2	2.6360	7.8e-1
6	2.6092	2.6360	-2.00e-2	7.84e-1	2.6099	-9.7e-4	2.6226	3.7e-1
7	2.6099	2.6226	-9.73e-4	3.72e-1	2.6099	-2.3e-5	2.6162	1.8e-1
8	2.6099	2.6162	-2.33e-5	1.83e-1	2.6099	-2.8e-7	2.6131	9.1e-2
9	2.6099	2.6131	-2.81e-7	9.10e-2	2.6099	-1.7e-9		

La modificación es mucho mejor que el método de Regula Falsi. Además, el número de iteraciones de la modificación debe ser menor o igual que el número de iteraciones del método de bisección. Pero para comparar equitativamente el método de bisección y la modificación de Regula Falsi, es necesario tener en cuenta el número de evaluaciones de $f(x)$.

En el método de bisección, en k iteraciones, el número de evaluaciones de f está dado por:

$$n_{\text{bisec}} = 2 + k_{\text{bisec}}\,.$$

En la modificación de Regula Falsi,

$$n_{\text{modif}} = 2 + 2\,k_{\text{modif}}\,.$$

4.7 Método de punto fijo

Los métodos vistos se aplican a la solución de la ecuación $f(x) = 0$. El método de punto fijo sirve para resolver la ecuación

$$g(x) = x. \tag{4.8}$$

Se busca un x^* tal que su imagen, por medio de la función g, sea el mismo x^*. Por tal motivo se dice que x^* es un punto fijo de la función g.

La aplicación del método es muy sencilla. A partir de un x_0 dado, se aplica varias veces la fórmula

$$x_{k+1} = g(x_k). \tag{4.9}$$

Se espera que la sucesión $\{x_k\}$ construida mediante (4.9) converja hacia x^*. En algunos casos el método, además de ser muy sencillo, es muy eficiente; en otros casos la eficiencia es muy pequeña; finalmente, en otros casos el método definitivamente no sirve.

Ejemplo 4.5. Resolver $x^3+x^2+6x+5 = 0$. Esta ecuación se puede escribir en la forma

$$x = -\frac{x^3 + x^2 + 5}{6}\,.$$

Aplicando el método de punto fijo a partir de $x_0 = -1$ se tiene:

$$\begin{array}{rcl} x_0 & = & \texttt{-1} \\ x_1 & = & \texttt{-0.833333} \\ x_2 & = & \texttt{-0.852623} \\ x_3 & = & \texttt{-0.851190} \\ x_4 & = & \texttt{-0.851303} \\ x_5 & = & \texttt{-0.851294} \\ x_6 & = & \texttt{-0.851295} \\ x_7 & = & \texttt{-0.851295} \\ x_8 & = & \texttt{-0.851295} \end{array}$$

Entonces se tiene una aproximación de una raíz, $x^* \approx -0.851295$. En este caso el método funcionó muy bien. Utilicemos ahora otra expresión para $x = g(x)$:

$$x = -\frac{x^3 + 6x + 5}{x}\,.$$

Aplicando el método de punto fijo a partir de $x_0 = -0.851$, muy buena aproximación de la raíz, se tiene:

$$\begin{array}{rcl} x_0 & = & \texttt{-0.8510} \\ x_1 & = & \texttt{-0.8488} \\ x_2 & = & \texttt{-0.8294} \\ x_3 & = & \texttt{-0.6599} \\ x_4 & = & \texttt{1.1415} \\ x_5 & = & \texttt{-11.6832} \\ x_6 & = & \texttt{-142.0691} \\ x_7 & = & \texttt{-2.0190e+4} \end{array}$$

En este caso se observa que, aun partiendo de una muy buena aproximación de la solución, no hay convergencia. $\diamond$

Antes de ver un resultado sobre convergencia del método de punto fijo, observemos su interpretación gráfica. La solución de $g(x) = x$ está determinada por el punto de corte, si lo hay, entre las gráficas $y = g(x)$ y $y = x$.

Después de dibujar las dos funciones, la construcción de los puntos x_1, x_2, x_3... se hace de la siguiente manera. Después de situar el valor x_0 sobre el eje x, para obtener el valor x_1, se busca verticalmente la curva $y = g(x)$. El punto obtenido tiene coordenadas $(x_0, g(x_0))$, o sea, (x_0, x_1). Para obtener $x_2 = g(x_1)$ es necesario inicialmente resituar x_1 sobre el eje x, para lo cual basta con buscar horizontalmente la recta $y = x$ para obtener el punto

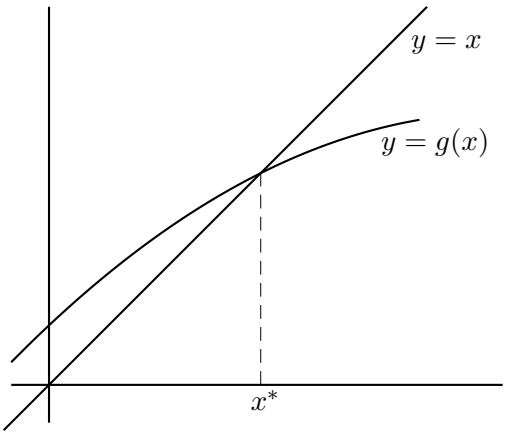


Figura 4.4: Punto fijo

(x_1, x_1). A partir de este punto se puede obtener x_2 buscando verticalmente la curva $y = g(x)$. Se tiene el punto $(x_1, g(x_1))$, o sea, (x_1, x_2). Con desplazamiento horizontal se obtiene (x_2, x_2). En resumen, se repite varias veces el siguiente procedimiento: *a partir de* (x_k, x_k) *buscar verticalmente en la curva* $y = g(x)$ *el punto* (x_k, x_{k+1}), *y a partir del punto obtenido buscar horizontalmente en la recta* $y = x$ *el punto* (x_{k+1}, x_{k+1}). Si el proceso converge, los puntos obtenidos tienden hacia el punto $(x^*, g(x^*)) = (x^*, x^*)$.

Las figuras 4.5 a 4.8 muestran gráficamente la utilización del método; en los dos primeros casos hay convergencia; en los otros dos no hay, aun si la aproximación inicial es bastante buena.

En seguida se presentan dos teoremas (demostración en [Atk78]) sobre la convergencia del método de punto fijo; el primero es más general y más preciso, el segundo es una simplificación del primero, de más fácil aplicación para ciertos problemas.

Teorema 4.2. *Sea* g *continuamente diferenciable en el intervalo* $[a, b]$, *tal que*

$$\begin{aligned} g([a,b]) &\subseteq [a,b], \\ |g'(x)| &< 1, \text{ para todo } x \in [a,b]. \end{aligned}$$

Entonces existe un único x^* *en* $[a, b]$ *solución de* $x = g(x)$ *y la iteración de punto fijo (4.9) converge a* x^* *para todo* $x_0 \in [a, b]$.

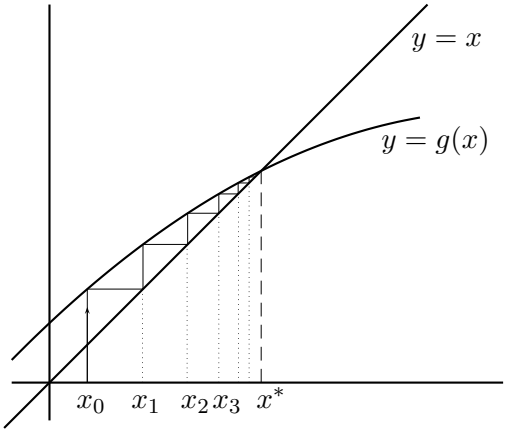


Figura 4.5: Método de punto fijo (a)

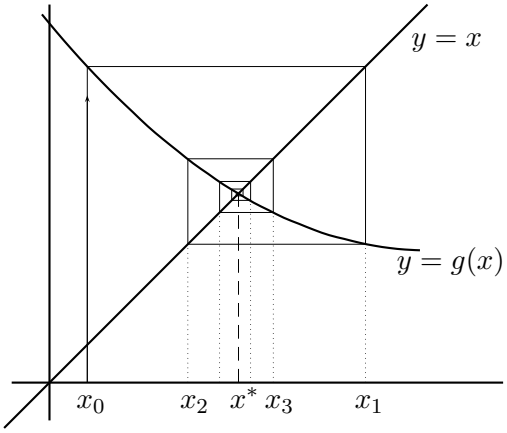


Figura 4.6: Método de punto fijo (b)

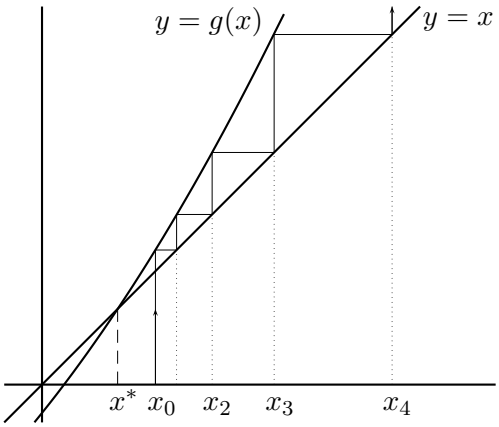


Figura 4.7: Método de punto fijo (c)

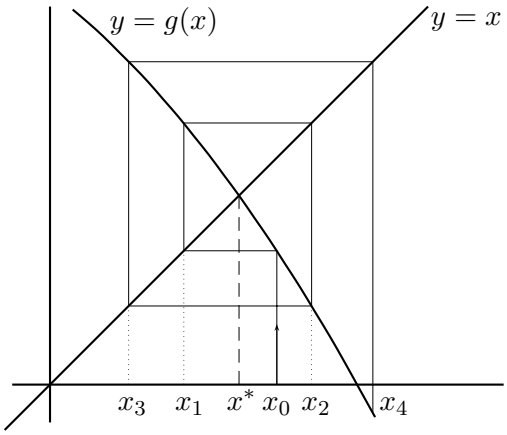


Figura 4.8: Método de punto fijo (d)

Teorema 4.3. *Sea x^* solución de $x = g(x)$, g continuamente diferenciable en un intervalo abierto I tal que $x^* \in I$, $|g'(x^*)| < 1$. Entonces la iteración de punto fijo (4.9) converge a x^* para todo x_0 suficientemente cerca de x^*.*

El caso ideal (la convergencia es más rápida) se tiene cuando $g'(x^*) \approx 0$.

En los dos ejemplos numéricos anteriores, para resolver $x^3+x^2+6x+5 = 0$, se tiene: $x = g(x) = -(x^3 + x^2 + 5)/6$, $g'(-0.8513) = -0.0786$. Si se considera $x = g(x) - (x^3 + 6x + 5)/x$, $g'(-0.8513) = 8.6020$. Estos resultados numéricos concuerdan con el último teorema.

Dos de los ejemplos gráficos anteriores muestran justamente que cuando $|g'(x^*)| < 1$ el método converge.

Ejemplo 4.6. Resolver $x^2 = 3$, o sea, calcular $\sqrt{3}$.

$$\begin{aligned} x^2 &= 3, \\ x^2 + x^2 &= x^2 + 3, \\ x &= \frac{x^2+3}{2x}, \\ x &= \frac{x + 3/x}{2}\,. \end{aligned}$$

$$\begin{aligned} x_0 &= 3 \\ x_1 &= 2 \\ x_2 &= 1.75000000000000 \\ x_3 &= 1.73214285714286 \\ x_4 &= 1.73205081001473 \\ x_5 &= 1.73205080756888 \\ x_6 &= 1.73205080756888 \end{aligned}$$

Se observa que la convergencia es bastante rápida. Este método es muy utilizado para calcular raíces cuadradas en calculadoras de bolsillo y computadores.

Aplicando el teorema 4.3 y teniendo en cuenta que $g'(x^*) = g'(\sqrt{3}) = 1/2 - 1.5/x^{*2} = 0$, se concluye rápidamente que si x^0 está suficientemente cerca de $\sqrt{3}$, entonces el método converge.

La aplicación del teorema 4.2 no es tan inmediata, pero se obtiene información más detallada. La solución está en el intervalo $[2, 3]$; consideremos

un intervalo aún más grande: $I = [1+\varepsilon, 4]$ con $0 < \varepsilon < 1$.

$$\begin{aligned} g(1) &= 2, \\ g(4) &= 2.375, \\ g'(x) &= \frac{1}{2} - \frac{3}{2x^2}, \\ g'(\sqrt{3}) &= 0, \\ g'(1) &= -1, \\ g'(4) &= \frac{13}{32}, \\ g''(x) &= \frac{3}{x^3}. \end{aligned}$$

Entonces $g''(x) > 0$ para todo x positivo. Luego $g'(x)$ es creciente para $x > 0$. Como $g'(1) = -1$, entonces $-1 < g'(1+\varepsilon)$. De nuevo por ser $g'(x)$ creciente, entonces $-1 < g'(x) \leq 13/32$ para todo $x \in I$. En resumen, $|g'(x)| < 1$ cuando $x \in I$.

Entre $1+\varepsilon$ y $\sqrt{3}$ el valor de $g'(x)$ es negativo. Entre $\sqrt{3}$ y 4 el valor de $g'(x)$ es positivo. Luego g decrece en $[1+\varepsilon, \sqrt{3}]$ y crece en $[\sqrt{3}, 4]$. Entonces $g([1+\varepsilon, \sqrt{3}]) = [g(1+\varepsilon), \sqrt{3}] \subseteq [2, \sqrt{3}]$ y $g([\sqrt{3}, 4]) = [\sqrt{3}, 2.375]$. En consecuencia $g(I) = [\sqrt{3}, 2.375] \subseteq I$. Entonces el método de punto fijo converge a $x^* = \sqrt{3}$ para cualquier $x_0 \in]1, 4]$. Este resultado se puede generalizar al intervalo $[1+\varepsilon, b]$ con $b > \sqrt{3}$.

Si se empieza con $x_0 = 1/2$, no se cumplen las condiciones del teorema; sin embargo, el método converge. $\diamond$

4.7.1 Modificación del método de punto fijo

La convergencia del método de punto fijo se puede tratar de mejorar retomando las ideas del método de la secante. Consideremos la ecuación $x = g(x)$ y los puntos $(x_i, g(x_i))$, $(x_j, g(x_j))$, sobre la gráfica de g. Estos puntos pueden provenir directamente o no del método de punto fijo. es decir, se puede tener que $x_{i+1} = g(x_i)$ y que $x_{j+1} = g(x_j)$, pero lo anterior no es obligatorio.

La idea consiste simplemente en obtener la ecuación de la recta que pasa por esos dos puntos y buscar la intersección con la recta $y = x$. La abcisa del punto dará un nuevo valor x_k.

$$
\begin{aligned}
y &= mx + b \\
m &= \frac{g(x_j) - g(x_i)}{x_j - x_i} \qquad (4.10) \\
g(x_i) &= mx_i + b \\
b &= g(x_i) - mx_i \qquad (4.11) \\
x_k &= mx_k + b \\
x_k &= \frac{b}{1-m}. \qquad (4.12)
\end{aligned}
$$

Ahora se usan los puntos $(x_j, g(x_j))$, $(x_k, g(x_k))$, para obtener un nuevo x_m, y así sucesivamente. Usualmente, $j = i + 1$ y $k = j + 1$.

4.7.2 Método de punto fijo y método de Newton

Supongamos que $c \neq 0$ es una constante y que x^* es solución de la ecuación $f(x) = 0$. Ésta se puede reescribir

$$
\begin{aligned}
0 &= cf(x), \\
x &= x + cf(x) = g(x). \qquad (4.13)
\end{aligned}
$$

Si se desea resolver esta ecuación por el método de punto fijo, la convergencia es más rápida cuando $g'(x^*) = 0$, o sea,

$$
\begin{aligned}
1 + cf'(x^*) &= 0, \\
c &= -\frac{1}{f'(x^*)}.
\end{aligned}
$$

Entonces al aplicar el método de punto fijo a (4.13), se tiene la fórmula

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x^*)}. \qquad (4.14)$$

Para aplicar esta fórmula se necesitaría conocer $f'(x^*)$ e implícitamente el valor de x^*, que es precisamente lo que se busca. La fórmula del método de Newton, (4.1), puede ser vista simplemente como la utilización de (4.14) reemplazando $f'(x^*)$ por la mejor aproximación conocida en ese momento: $f'(x_k)$.

4.8 Método de Newton en $\mathbb{R}^n$

Consideremos ahora un sistema de n ecuaciones con n incógnitas; por ejemplo,

$$\begin{aligned} x_1^2 + x_1x_2 + x_3 - 3 &= 0 \\ 2x_1 + 3x_2x_3 - 5 &= 0 \\ (x_1 + x_2 + x_3)^2 - 10x_3 + 1 &= 0. \end{aligned} \tag{4.15}$$

Este sistema no se puede escribir en la forma matricial $Ax = b$; entonces no se puede resolver por los métodos usuales para sistemas de ecuaciones lineales. Lo que se hace, como en el método de Newton en $\mathbb{R}$, es utilizar aproximaciones de primer orden (llamadas también aproximaciones lineales). Esto es simplemente la generalización de la aproximación por una recta.

Un sistema de n ecuaciones con n incógnitas se puede escribir de la forma

$$\begin{aligned} F_1(x_1, x_2, ..., x_n) &= 0 \\ F_2(x_1, x_2, ..., x_n) &= 0 \\ &\vdots \\ F_n(x_1, x_2, ..., x_n) &= 0, \end{aligned}$$

donde cada F_i es una función de n variables con valor real, o sea, $F_i : \mathbb{R}^n \to \mathbb{R}$. Denotemos $x = (x_1, x_2, ..., x_n)$ y

$$F(x) = \begin{bmatrix} F_1(x) \\ F_2(x) \\ \vdots \\ F_n(x) \end{bmatrix}.$$

Así F es una función de variable vectorial y valor vectorial, $F : \mathbb{R}^n \to \mathbb{R}^n$, y el problema se escribe de manera muy compacta:

$$F(x) = 0. \tag{4.16}$$

Este libro está dirigido principalmente a estudiantes de segundo semestre, quienes todavía no conocen el cálculo en varias variables, entonces no habrá una deducción (ni formal ni intuitiva) del método, simplemente se verá como una generalización del método en $\mathbb{R}$.

4.8.1 Matriz jacobiana

La matriz jacobiana de la función $F : \mathbb{R}^n \to \mathbb{R}^n$, denotada por $J_F(\bar{x})$ o por $F'(\bar{x})$, es una matriz de tamaño $n \times n$, en la que en la i-ésima fila están las n derivadas parciales de F_i,

$$J_F(x) = F'(x) = \begin{bmatrix} \dfrac{\partial F_1}{\partial x_1}(x) & \dfrac{\partial F_1}{\partial x_2}(x) & \cdots & \dfrac{\partial F_1}{\partial x_n}(x) \\ \dfrac{\partial F_2}{\partial x_1}(x) & \dfrac{\partial F_2}{\partial x_2}(x) & \cdots & \dfrac{\partial F_2}{\partial x_n}(x) \\ \vdots & & \ddots & \\ \dfrac{\partial F_n}{\partial x_1}(x) & \dfrac{\partial F_n}{\partial x_2}(x) & \cdots & \dfrac{\partial F_n}{\partial x_n}(x) \end{bmatrix}$$

Para las ecuaciones (4.15), escritas en la forma $F(x) = 0$,

$$F'(x) = \begin{bmatrix} 2x_1 + x_2 & x_1 & 1 \\ 2 & 3x_3 & 3x_2 \\ 2(x_1 + x_2 + x_3) & 2(x_1 + x_2 + x_3) & 2(x_1 + x_2 + x_3) - 10 \end{bmatrix}$$

$$F'(2, -3, 4) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 12 & -9 \\ 6 & 6 & -4 \end{bmatrix}.$$

4.8.2 Fórmula de Newton en $\mathbb{R}^n$

La fórmula del método de Newton en $\mathbb{R}$,

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

se puede reescribir con superíndices en lugar de subíndices:

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}.$$

De nuevo, es simplemente otra forma de escribir

$$x^{k+1} = x^k - f'(x^k)^{-1} f(x^k).$$

Esta expresión sí se puede generalizar

$$x^{k+1} = x^k - F'(x^k)^{-1} F(x^k). \tag{4.17}$$

Su interpretación, muy natural, aparece a continuación. Sea x^*, un vector de n componentes, solución del sistema (4.16). Dependiendo de la conveniencia se podrá escribir

$$x^* = (x_1^*, x_2^*, ..., x_n^*) \qquad \text{o} \qquad x^* = \begin{bmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_n^* \end{bmatrix}.$$

El método empieza con un vector $x^0 = (x_1^0, x_2^0, ..., x_n^0)$, aproximación inicial de la solución x^*. Mediante (4.17) se construye una sucesión de vectores $\{x^k = (x_1^k, x_2^k, ..., x_n^k)\}$ con el deseo de que $x^k \to x^*$. En palabras, el vector x^{k+1} es igual al vector x^k menos el producto de la inversa de la matriz jacobiana $F'(x^k)$ y el vector $F(x^k)$. Para evitar el cálculo de una inversa, la fórmula se puede reescribir

$$\begin{aligned} d^k &= -F'(x^k)^{-1} F(x^k) \\ x^{k+1} &= x^k + d^k. \end{aligned}$$

Premultiplicando por $F'(x^k)$

$$\begin{aligned} F'(x^k)\, d^k &= -F'(x^k) F'(x^k)^{-1} F(x^k), \\ F'(x^k)\, d^k &= -F(x^k). \end{aligned}$$

En esta última expresión se conoce (o se puede calcular) la matriz $F'(x^k)$. También se conoce el vector $F(x^k)$. O sea, simplemente se tiene un sistema de ecuaciones lineales. La solución de este sistema es el vector d^k. Entonces las fórmulas para el método de Newton son:

$$\begin{aligned} \text{resolver} \quad F'(x^k)\, d^k &= -F(x^k), \\ x^{k+1} &= x^k + d^k. \end{aligned} \tag{4.18}$$

Ejemplo 4.7. Resolver el sistema

$$\begin{aligned} x_1^2 + x_1x_2 + x_3 - 3 &= 0 \\ 2x_1 + 3x_2x_3 - 5 &= 0 \\ (x_1 + x_2 + x_3)^2 - 10x_3 + 1 &= 0 \end{aligned}$$

a partir de $x^0 = (2, -3, 4)$.

$$F(x^0) = \begin{bmatrix} -1 \\ -37 \\ -30 \end{bmatrix} \quad , \quad F'(x^0) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 12 & -9 \\ 6 & 6 & -4 \end{bmatrix}$$

$$\text{resolver } \begin{bmatrix} 1 & 2 & 1 \\ 2 & 12 & -9 \\ 6 & 6 & -4 \end{bmatrix} \begin{bmatrix} d_1^0 \\ d_2^0 \\ d_3^0 \end{bmatrix} = - \begin{bmatrix} -1 \\ -37 \\ -30 \end{bmatrix} \quad , \quad d^0 = \begin{bmatrix} 2.5753 \\ 0.5890 \\ -2.7534 \end{bmatrix}$$

$$x^1 = \begin{bmatrix} 2 \\ -3 \\ 4 \end{bmatrix} + \begin{bmatrix} 2.5753 \\ 0.5890 \\ -2.7534 \end{bmatrix} = \begin{bmatrix} 4.5753 \\ -2.4110 \\ 1.2466 \end{bmatrix}$$

$$F(x^1) = \begin{bmatrix} 8.1494 \\ -4.8656 \\ 0.1689 \end{bmatrix} \quad , \quad F'(x^1) = \begin{bmatrix} 6.7397 & 4.5753 & 1.0000 \\ 2.0000 & 3.7397 & -7.2329 \\ 6.8219 & 6.8219 & -3.1781 \end{bmatrix}$$

$$\begin{bmatrix} 6.7397 & 4.5753 & 1.0000 \\ 2.0000 & 3.7397 & -7.2329 \\ 6.8219 & 6.8219 & -3.1781 \end{bmatrix} \begin{bmatrix} d_1^1 \\ d_2^1 \\ d_3^1 \end{bmatrix} = - \begin{bmatrix} 8.1494 \\ -4.8656 \\ 0.1689 \end{bmatrix}, \; d^1 = \begin{bmatrix} -4.4433 \\ 4.6537 \\ 0.5048 \end{bmatrix}$$

$$x^2 = \begin{bmatrix} 4.5753 \\ -2.4110 \\ 1.2466 \end{bmatrix} + \begin{bmatrix} -4.4433 \\ 4.6537 \\ 0.5048 \end{bmatrix} = \begin{bmatrix} 0.1321 \\ 2.2428 \\ 1.7514 \end{bmatrix}$$

A continuación se presentan los resultados de $F(x^k)$, $F'(x^k)$, d^k, x^{k+1}. $k = 2$

$$\begin{bmatrix} -0.9350 \\ 7.0481 \\ 0.5116 \end{bmatrix}, \begin{bmatrix} 2.5069 & 0.1321 & 1.0000 \\ 2.0000 & 5.2542 & 6.7283 \\ 8.2524 & 8.2524 & -1.7476 \end{bmatrix}, \begin{bmatrix} 0.6513 \\ -0.8376 \\ -0.5870 \end{bmatrix}, \begin{bmatrix} 0.7833 \\ 1.4052 \\ 1.1644 \end{bmatrix}$$

$k = 3$

$$\begin{bmatrix} -0.1213 \\ 1.4751 \\ 0.5981 \end{bmatrix}, \begin{bmatrix} 2.9718 & 0.7833 & 1.0000 \\ 2.0000 & 3.4931 & 4.2156 \\ 6.7057 & 6.7057 & -3.2943 \end{bmatrix}, \begin{bmatrix} 0.1824 \\ -0.3454 \\ -0.1502 \end{bmatrix}, \begin{bmatrix} 0.9658 \\ 1.0598 \\ 1.0141 \end{bmatrix}$$

$k = 4$

$$\begin{bmatrix} -0.0297 \\ 0.1557 \\ 0.0981 \end{bmatrix}, \quad \begin{bmatrix} 2.9913 & 0.9658 & 1.0000 \\ 2.0000 & 3.0424 & 3.1793 \\ 6.0793 & 6.0793 & -3.9207 \end{bmatrix}, \quad \begin{bmatrix} 0.0335 \\ -0.0587 \\ -0.0139 \end{bmatrix}, \quad \begin{bmatrix} 0.9993 \\ 1.0011 \\ 1.0002 \end{bmatrix}$$

$k = 5$

$$\begin{bmatrix} -0.0008 \\ 0.0025 \\ 0.0015 \end{bmatrix}, \quad \begin{bmatrix} 2.9997 & 0.9993 & 1.0000 \\ 2.0000 & 3.0006 & 3.0033 \\ 6.0012 & 6.0012 & -3.9988 \end{bmatrix}, \quad \begin{bmatrix} 0.0007 \\ -0.0011 \\ -0.0002 \end{bmatrix}, \quad \begin{bmatrix} 1.0000 \\ 1.0000 \\ 1.0000 \end{bmatrix}$$

$$F(x^6) \approx \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \text{luego} \quad x^* \approx \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \ \diamond$$

4.9 Método de Muller

Este método sirve para hallar raíces reales o complejas de polinomios. Sea $p(x)$ un polinomio real (con coeficientes reales), de grado n, es decir,

$$p(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n, \quad a_i \in \mathbb{R}, \ i = 0, 1, ..., n, \quad a_n \neq 0.$$

En general no se puede garantizar que $p(x)$ tenga raíces reales. Sin embargo (teorema fundamental del Álgebra) se puede garantizar que tiene n raíces complejas (algunas de ellas pueden ser reales). De manera más precisa, existen $r_1, r_2, ..., r_n \in \mathbb{C}$ tales que

$$p(r_i) = 0, \quad i = 1, 2, ..., n.$$

El polinomio p se puede expresar en función de sus raíces:

$$p(x) = a_n(x - r_1)(x - r_2) \cdots (x - r_n).$$

Las raíces complejas, no reales, siempre vienen por parejas, es decir si $r = a + ib$, $b \neq 0$, es una raíz entonces $\bar{r} = a - ib$, el conjugado de r, también es raíz. Esto garantiza que los polinomios de grado impar tienen por lo menos una raíz real. Para los polinomios de grado par, el número de raíces reales es par y el número de raíces estrictamente complejas también es par. Así un polinomio de grado par puede tener cero raíces reales.

Para las raíces complejas $(x - r)(x - \bar{r})$ divide a $p(x)$.

$(x-r)(x-\bar{r}) = (x-a-ib)(x-a+ib) = (x-a)^2+b^2 = x^2-2ax+(a^2+b^2)$.

O sea, se tiene un polinomio real de grado 2 que divide a $p(x)$.

Si $q(x)$ divide a $p(x)$, entonces existe un polinomio $s(x)$ tal que

$$\begin{aligned} p(x) &= q(x)s(x), \\ \text{grado}(p) &= \text{grado}(q) + \text{grado}(s). \end{aligned}$$

Entonces para sequir obteniendo las raíces de $p(x)$ basta con obtener las raíces de $s(x)$, polinomio más sencillo.

Si se halla una raíz real r entonces $q(x) = (x-r)$ divide a $p(x)$. Si se obtiene una raíz compleja $r = a + ib$, entonces $q(x) = x^2 - 2ax + (a^2 + b^2)$ divide a $p(x)$. Este proceso de obtener un polinomio de grado menor cuyas raíces sean raíces del polinomio inicial se llama **deflación**.

En el método de la secante, dados dos valores x_0 y x_1 se busca la recta que pasa por los puntos $(x_0, f(x_0))$, $(x_1, f(x_1))$; el siguiente valor x_2 está dado por el punto donde la recta corta el eje x.

En el método de Muller, en lugar de una recta, se utiliza una parábola. Dados tres valores x_0, x_1 y x_2, se construye la parábola $P(x)$ que pasa por los puntos $(x_0, f(x_0))$, $(x_1, f(x_1))$ y $(x_2, f(x_2))$; el siguiente valor x_3 está dado por el (un) punto tal que $P(x_3) = 0$.

La parábola se puede escribir de la forma $P(x) = a(x - x_2)^2 + b(x - x_2) + c$. Entonces hay tres condiciones que permiten calcular a, b y c:

$$\begin{aligned} f(x_0) &= a(x_0 - x_2)^2 + b(x_0 - x_2) + c, \\ f(x_1) &= a(x_1 - x_2)^2 + b(x_1 - x_2) + c, \\ f(x_2) &= c. \end{aligned}$$

Después de algunos cálculos se obtiene

$$\begin{aligned} d &= (x_0 - x_1)(x_0 - x_2)(x_1 - x_2), \\ a &= \frac{-(x_0 - x_2)(f(x_1) - f(x_2)) + (x_1 - x_2)(f(x_0) - f(x_2)}{d}, \\ b &= \frac{(x_0 - x_2)^2(f(x_1) - f(x_2)) - (x_1 - x_2)^2(f(x_0) - f(x_2)}{d}, \\ c &= f(x_2). \end{aligned} \tag{4.19}$$

Entonces

$$x_3 - x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Para reducir los errores de redondeo se "racionaliza" el numerador y se escoge el signo buscando que el denominador resultante sea grande (en valor absoluto)

$$\begin{aligned} D &= b^2 - 4ac, \\ R &= \sqrt{D} \\ x_3 - x_2 &= \frac{-b \pm R}{2a} \ \frac{-b \mp R}{-b \mp R} \\ x_3 - x_2 &= \frac{b^2 - R^2}{2a\big(-b \mp R\big)} = \frac{b^2 - b^2 + 4ac}{2a\big(-b \mp R\big)} = \frac{2c}{-b \mp R} \\ x_3 - x_2 &= -\frac{2c}{b \pm R} \\ x_3 &= x_2 - \frac{2c}{b + \text{signo}(b)R} \end{aligned} \tag{4.20}$$

En la siguiente iteración se obtiene la parábola utilizando x_1, x_2 y x_3 para obtener x_4.

Si en una iteración

$$D = b^2 - 4ac < 0$$

es necesario utilizar, a partir de ahí, aritmética compleja (Scilab lo hace automáticamente). Eso hace que los siguientes valores a, b y c no sean necesariamente reales. Muy posiblemente $b^2 - 4ac$ tampoco es real. Para utilizar (4.20) es necesario obtener la raíz cuadradada de un complejo.

Sean z un complejo, θ el ángulo (en radianes) formado con el eje real ("eje x"), llamado con frecuencia argumento de z, y ρ la norma o valor absoluto de z. La dos raíces cuadradas de z son:

$$\begin{aligned} \sqrt{z} = \zeta_1 &= \sqrt{\rho}\,\big(\cos(\theta/2) + i\,\text{sen}(\theta/2)\big), \\ \zeta_2 &= -\zeta_1. \end{aligned}$$

Ejemplo 4.8. Sea $z = 12 + 16i$. Entonces

$$\begin{aligned} \rho &= 20, \\ \theta &= \tan^{-1}(16/12) = 0.927295, \\ \zeta_1 &= \sqrt{20}\,\big(\cos(0.927295/2) + i\ \text{sen}(0.927295/2)\big) = 4 + 2i, \\ \zeta_2 &= -4 - 2i. \quad \diamond \end{aligned}$$

Cuando b no es real, es necesario modificar ligeramente (4.20). Se escoge el signo para que el denominador tenga máxima norma:

$$\begin{aligned} D &= b^2 - 4ac \\ R &= \sqrt{D} \\ \delta &= \begin{cases} b+R & \text{si} \quad |b+R| \geq |b-R| \\ b-R & \text{si} \quad |b+R| < |b-R| \end{cases} \\ x_3 &= x_2 - \frac{2c}{\delta} \ . \end{aligned} \tag{4.21}$$

Ejemplo 4.9. Hallar las raíces de $p(x) = 2x^5 + x^4 + 4x^3 + 19x^2 - 18x + 40$ partiendo de $x_0 = 0$, $x_1 = 0.5$, $x_2 = 1$.

$$\begin{aligned} f(x_0) &= 40 \\ f(x_1) &= 36.375 \\ f(x_2) &= 48 \\ d &= -0.25 \\ a &= 30.5 \\ b &= 38.5 \\ c &= 48 \\ D &= -4373.75 \end{aligned}$$

Hay que utilizar aritmética compleja

$$\begin{aligned} R &= 66.134333i \\ \delta &= 38.5 + 66.134333i \\ x_3 &= 0.368852 + 1.084169i \\ f(x_3) &= 12.981325 - 9.579946i \end{aligned}$$

Ahora utilizamos x_1, x_2 y x_3

$$\begin{aligned}
d &= 0.546325 + 0.413228i \\
a &= 27.161207 + 11.293018i \\
b &= -21.941945 + 50.286087i \\
c &= 12.981325 - 9.579946i \\
D &= -3890.341507 - 1752.330850i \\
R &= 13.719321 - 63.863615i \\
\delta &= -35.661266 + 114.149702i \\
x_4 &= 0.586513 + 1.243614i \\
f(x_4) &= 3.760763 - 6.548104i \\
&\vdots \\
x_5 &= 0.758640 + 1.246582i \\
f(x_5) &= -2.013839 - 1.490220i \\
x_6 &= 0.748694 + 1.196892i \\
f(x_6) &= 0.123017 + 0.025843i \\
x_7 &= 0.750002 + 1.198942i \\
f(x_7) &= 0.000535 + 0.000636i \\
x_8 &= 0.750000 + 1.198958i \\
f(x_8) &= 0
\end{aligned}$$

Ahora se construye el polinomio $q(x) = (x - r)(x - \bar{r})$. Para $r = 0.75 + 1.198958i$ se tiene $q(x) = x^2 - 1.5x + 2$.

$$\frac{2x^5 + x^4 + 4x^3 + 19x^2 - 18x + 40}{x^2 - 1.5x + 2} = 2x^3 + 4x^2 + 6x^2 + 20.$$

Ahora se trabaja con $p(x) = 2x^3 + 4x^2 + 6x^2 + 20$. Sean $x_0 = -3$, $x_1 = -2.5$ y $x_2 = -2$. También se hubiera podido volver a utilizar $x_0 = 0$, $x_1 = 0.5$ y

$x_2 = 1$.

$$\begin{aligned} f(x_0) &= -16 \\ f(x_1) &= -1.25 \\ f(x_2) &= 8 \\ d &= -0.25 \\ a &= -11 \\ b &= 13 \\ c &= 8 \\ D &= 521 \\ R &= 22.825424 \\ \delta &= 35.825424 \\ x_3 &= -2.446610 \\ f(x_3) &= -0.026391 \end{aligned}$$

Ahora utilizamos x_1, x_2 y x_3

$$\begin{aligned} d &= 0.011922 \\ a &= -9.893220 \\ b &= 22.390216 \\ c &= -0.026391 \\ D &= 500.277428 \\ R &= 22.366882 \\ \delta &= 44.757098 \\ x_4 &= -2.445431 \\ f(x_4) &= -0.000057 \\ &\vdots \\ x_5 &= -2.445428 \\ f(x_5) &= 0 \end{aligned}$$

Para $r = -2.445428$ se tiene $q(x) = x + 2.445428$.

$$\frac{2x^3 + 4x^2 + 6x^2 + 20}{x + 2.445428} = 2x^2 - 0.890857x + 8.178526.$$

Ahora se trabaja con $p(x) = 2x^2 - 0.890857x + 8.178526$. Sus raíces son $0.2227142 + 2.009891i$ y $0.2227142 - 2.009891i$. En resumen, las 5 raíces de $p(x)$ son:

$$0.75 + 1.1989579i$$
$$0.75 - 1.1989579i$$
$$-2.445428$$
$$0.222714 + 2.009891i$$
$$0.222714 - 2.009891i. \ \diamond$$

El método de Muller tiene orden de convergencia no inferior a 1.84... Este valor proviene de la raíz mas grande de $\mu^3 - \mu^2 - \mu - 1 = 0$. Esto hace que sea un poco menos rápido que el método de Newton (orden 2) pero más rápido que el método de la secante (orden 1.68).

El método no tiene sentido si hay valores iguales (o muy parecidos) entre x_0, x_1 y x_2. Además esto haría que no se pueda calcular a ni b. Tampoco funciona si los valores $f(x_0)$, $f(x_1)$ y $f(x_2)$ son iguales o muy parecidos. En este caso $P(x)$ es una línea recta horizontal y no se puede calcular x_3 ya que $a = 0$, $b = 0$ y, principalmente, $\delta = b \pm R = 0$.

MÉTODO DE MULLER PARA UNA RAÍZ

datos: p, x_0, x_1, x_2, ε_f, ε_0, maxit
aritmética = real
$f_0 = p(x_0)$, $f_1 = p(x_1)$, $f_2 = p(x_2)$
`info`= 0
para $k = 1, ..., \text{maxit}$
  si $|f_2| \leq \varepsilon_f$ **ent** $r = x_2$, `info`= 1, **parar**
  $d = (x_0 - x_1)(x_0 - x_2)(x_1 - x_2)$
  si $|d| \leq \varepsilon_0$ **ent parar**
  calcular a, b y c según (4.19)
  $D = b^2 - 4ac$
  si aritmética=real y $D < 0$ **ent** aritmética=compleja
  $R = \sqrt{D}$
  $\delta_1 = b + R, \quad \delta_2 = b - R$
  si $|\delta_1| \geq |\delta_2|$ **ent** $\delta = \delta_1$, **sino** $\delta = \delta_2$
  si $|\delta| \leq \varepsilon_0$ **ent parar**
  $x_3 = x_2 - 2c/\delta$
  $x_0 = x_1$, $x_1 = x_2$, $x_2 = x_3$, $f_0 = f_1$, $f_1 = f_2$
  $f_2 = p(x_2)$
fin-para k

Si el algoritmo anterior acaba normalmente, `info` valdrá 1 y r será una raíz, real o compleja.

MÉTODO DE MULLER

datos: p, x_0, ε_f, ε_0, maxit
$r = x_0, \quad h = 0.5$
mientras grado$(p) \geq 3$
  $x_0 = r, \;\; x_1 = x_0 + h, \;\; x_2 = x_1 + h$
  $(r, \; info) = Muller1(p, x_o, x_1, x_2, \varepsilon_f, \varepsilon_0, maxit)$
  si $info = 0$, **ent parar**
  si $|imag(r)| \leq \varepsilon_0$ **ent** $q(x) = (x - r)$
  sino $q(x) = (x - r)(x - \bar{r})$
  $p(x) = p(x)/q(x)$
fin-mientras
calcular raíces de p (de grado no superior a 2)

Si se espera que el número de raíces reales sea pequeño, comparado con el de raíces complejas, se puede trabajar todo el tiempo con aritmética compleja.

4.10 Método de Bairstow

Sirve para hallar las raíces reales o complejas de un polinomio de grado mayor o igual a 4, mediante la obtención de los factores cuadráticos "mónicos" del polinomio. Cuando es de grado 3, se halla una raíz real por el método de Newton, y después de la deflación se calculan las 2 raíces del polinomio cuadrático resultante.

Sea

$$p(x) = \alpha_n x^n + \alpha_{n-1} x^{n-1} + \alpha_{n-2} x^{n-2} + ... + \alpha_1 x + \alpha_0$$

reescrito como

$$p(x) = u_1 x^n + u_2 x^{n-1} + u_3 x^{n-2} + ... + u_n x + u_{n+1} \tag{4.22}$$

Se desea encontrar $x^2 - dx - e$ divisor de p. Cuando se hace la división entre p y un polinomio cuadrático cualquiera, se obtiene un residuo $r(x) = Rx + S$. Entonces se buscan valores de d y e tales que $r(x) = 0$, es decir, $R = 0$ y $S = 0$. Los valores R y S dependen de d y e, o sea, $R = R(d, e)$ y $S = S(d, e)$

Tenemos dos ecuaciones con dos incógnitas,

$$R(d, e) = 0$$
$$S(d, e) = 0$$

Sea

$$q(x) = \beta_{n-2} x^{n-2} + \beta_{n-3} x^{n-3} + ... + \beta_1 x + \beta_0$$

reescrito como

$$q(x) = v_1 x^{n-2} + v_2 x^{n-3} + ... + v_{n-2} x + v_{n-1}$$

el cociente. Entonces

$$p(x) = q(x)(x^2 - dx - e) + Rx + S.$$

Es decir,

$$u_1 x^n + u_2 x^{n-1} + ... + u_n x + u_{n+1} = (v_1 x^{n-2} + v_2 x^{n-3} + ... + v_{n-2} x + v_{n-1})(x^2 - dx - e) + R$$

$$
\begin{aligned}
u_1 &= v_1 \\
u_2 &= v_2 - dv_1 \\
u_3 &= v_3 - dv_2 - ev_1 \\
u_4 &= v_4 - dv_3 - ev_2
\end{aligned}
$$

$$
u_i = v_i - dv_{i-1} - ev_{i-2}
$$

$$
\begin{aligned}
u_{n-1} &= v_{n-1} - dv_{n-2} - ev_{n-3} \\
u_n &= -dv_{n-1} - ev_{n-2} + R \\
u_{n+1} &= -ev_{n-1} + S
\end{aligned}
$$

Para facilitar las fórmulas es útil introducir dos coeficientes adicionales, v_n y v_{n+1}, que no influyen sobre q, definidos por

$$
\begin{aligned}
v_n &= R \\
v_{n+1} &= S + dv_n
\end{aligned}
$$

Entonces:

$$
\begin{aligned}
u_n &= v_n - dv_{n-1} - ev_{n-2} \\
u_{n+1} &= dv_n - dv_n - ev_{n-1} + S \\
\text{o sea,} \quad u_{n+1} &= v_{n+1} - dv_n - ev_{n-1}
\end{aligned}
$$

Las igualdades quedan:

$$
\begin{aligned}
u_1 &= v_1 \\
u_2 &= v_2 - dv_1 \\
u_i &= v_i - dv_{i-1} - ev_{i-2}, \qquad i = 3, ..., n+1.
\end{aligned}
$$

Las fórmulas para calcular los v_i son

$$\begin{aligned} v_1 &= u_1 \\ v_2 &= u_2 + dv_1 \\ v_i &= u_i + dv_{i-1} + ev_{i-2}, \qquad i = 3, ..., n+1. \end{aligned} \tag{4.23}$$

Una vez obtenidos los v_i, entonces

$$\begin{aligned} R &= v_n \\ S &= v_{n+1} - dv_n \end{aligned}$$

	u_1	u_2	u_3	u_4	$\cdots$	u_{n+1}
d		dv_1	dv_2	dv_3	$\cdots$	dv_n
e			ev_1	ev_2	$\cdots$	ev_{n-1}
	$v_1 = u_1$	$v_2 = \Sigma$	$v_3 = \Sigma$	$v_4 = \Sigma$		$v_{n+1} = \Sigma$

$$R = v_n, \quad S = v_{n+1} - dv_n$$

	4	5	1	0	-1	2
2		8	26	30	-18	-128
-3			-12	-39	-45	27
	4	13	15	-9	-64	-99

$$R = -64, \quad S = -99 - 2 \times (-64) = 29$$

El objetivo inicial era buscar $R = 0$ y $S = 0$. Esto se obtiene si $v_n = 0$ y $v_{n+1} = 0$. O sea, ahora lo que se desea es encontrar d y e tales que

$$\begin{aligned} v_n(d, e) &= 0 \\ v_{n+1}(d, e) &= 0 \end{aligned}$$

Al aplicar el método de Newton se tiene:

$$\text{resolver el sistema} \quad J \begin{bmatrix} \Delta d^k \\ \Delta e^k \end{bmatrix} = - \begin{bmatrix} v_n(d^k, e^k) \\ v_{n+1}(d^k, e^k) \end{bmatrix} \tag{4.24}$$

$$\begin{bmatrix} d^{k+1} \\ e^{k+1} \end{bmatrix} = \begin{bmatrix} d^k \\ e^k \end{bmatrix} + \begin{bmatrix} \Delta d^k \\ \Delta e^k \end{bmatrix} \tag{4.25}$$

donde J es la matriz jacobiana

$$J = \begin{bmatrix} \dfrac{\partial v_n}{\partial d}(d^k, e^k) & \dfrac{\partial v_n}{\partial e}(d^k, e^k) \\ \dfrac{\partial v_{n+1}}{\partial d}(d^k, e^k) & \dfrac{\partial v_{n+1}}{\partial e}(d^k, e^k) \end{bmatrix}.$$

Cálculo de las derivadas parciales:

$$\begin{aligned} \frac{\partial v_1}{\partial d} &= 0 \\ \frac{\partial v_2}{\partial d} &= v_1 \\ \frac{\partial v_i}{\partial d} &= v_{i-1} + d\frac{\partial v_{i-1}}{\partial d} + e\frac{\partial v_{i-2}}{\partial d} \end{aligned}$$

$$\begin{aligned} \frac{\partial v_1}{\partial e} &= 0 \\ \frac{\partial v_2}{\partial e} &= 0 \\ \frac{\partial v_i}{\partial e} &= d\frac{\partial v_{i-1}}{\partial e} + v_{i-2} + e\frac{\partial v_{i-2}}{\partial e} \\ \frac{\partial v_i}{\partial e} &= v_{i-2} + d\frac{\partial v_{i-1}}{\partial e} + e\frac{\partial v_{i-2}}{\partial e} \end{aligned}$$

Explicitando las derivadas parciales con respecto a d se tiene

$$\begin{aligned}
\frac{\partial v_1}{\partial d} &= 0 \\
\frac{\partial v_2}{\partial d} &= v_1 \\
\frac{\partial v_3}{\partial d} &= v_2 + d\frac{\partial v_2}{\partial d} + e\frac{\partial v_1}{\partial d} \\
\frac{\partial v_3}{\partial d} &= v_2 + d\frac{\partial v_2}{\partial d} \\
\frac{\partial v_4}{\partial d} &= v_3 + d\frac{\partial v_3}{\partial d} + e\frac{\partial v_2}{\partial d} \\
\frac{\partial v_i}{\partial d} &= v_{i-1} + d\frac{\partial v_{i-1}}{\partial d} + e\frac{\partial v_{i-2}}{\partial d}
\end{aligned}$$

Sea

$$\begin{aligned}
w_1 &= v_1 \\
w_2 &= v_2 + dw_1 \\
w_i &= v_i + dw_{i-1} + ew_{i-2}, \qquad i = 3, ..., n.
\end{aligned} \tag{4.26}$$

Es importante observar que estas fórmulas son análogas a las de la división sintética doble, que permiten obtener, a partir de los valores u_i, los valores v_i.

La derivar se tiene:

$$\begin{aligned}
\frac{\partial v_1}{\partial d} &= 0 \\
\frac{\partial v_2}{\partial d} &= w_1 \\
\frac{\partial v_3}{\partial d} &= w_2 \\
\frac{\partial v_i}{\partial d} &= w_{i-1}
\end{aligned}$$

Explicitando las derivadas parciales con respecto a e se tiene

$$\begin{aligned}
\frac{\partial v_1}{\partial e} &= 0 \\
\frac{\partial v_2}{\partial e} &= 0 \\
\frac{\partial v_3}{\partial e} &= v_1 \\
\frac{\partial v_4}{\partial e} &= v_2 + dv_1 \\
\frac{\partial v_5}{\partial e} &= v_3 + d\frac{\partial v_4}{\partial e} + e\frac{\partial v_3}{\partial e}
\end{aligned}$$

Utilizando de nuevo los w_i

$$\begin{aligned}
\frac{\partial v_1}{\partial e} &= 0 \\
\frac{\partial v_2}{\partial e} &= 0 \\
\frac{\partial v_3}{\partial e} &= w_1 \\
\frac{\partial v_4}{\partial e} &= w_2 \\
\frac{\partial v_5}{\partial e} &= w_3 \\
\frac{\partial v_i}{\partial e} &= w_{i-2}
\end{aligned}$$

Entonces

$$\begin{aligned}
\frac{\partial v_n}{\partial d} &= w_{n-1} \\
\frac{\partial v_n}{\partial e} &= w_{n-2} \\
\frac{\partial v_{n+1}}{\partial d} &= w_n \\
\frac{\partial v_{n+1}}{\partial e} &= w_{n-1}
\end{aligned}$$

Es decir, la matriz jacobiana es simplemente

$$J = \begin{bmatrix} w_{n-1} & w_{n-2} \\ w_n & w_{n-1} \end{bmatrix}. \tag{4.27}$$

MÉTODO DE BAIRSTOW

datos: u_1, u_2, ..., u_{n+1} (4.22), d^0, e^0, ε, `MAXIT`
para $k = 0, ...,$`MAXIT`
    calcular v_1, v_2, ..., v_{n+1} según (4.23)
    si $||\,(\,v_n,\, v_{n+1}\,)\,|| \leq \varepsilon$, **ent parar**
    calcular w_1, w_2, ..., w_n según (4.26)
    construir J según (4.27)
    resolver el sistema (4.24)
    obtener d^{k+1} y e^{k+1} según (4.25)
fin-para k

Si el agoritmo acaba de la manera esperada, $||\,(\,v_n,\, v_{n+1}\,)\,|| \leq \varepsilon$, entonces los últimos valores d y e hacen que $x^2 - dx - e$ divida "exactamente" a $p(x)$. El cociente será justamente $q(x) = v_1 x^{n-2} + v_2 x^{n-3} + ... + v_{n-2}x + v_{n-1}$. Así, las dos raíces de $x^2 - dx - e$ son también raíces de $p(x)$. Si el grado de q es superior a dos, entonces se puede recomenzar el proceso con $q(x)$.

El método de Bairstow es, en el fondo, el método de Newton en $\mathbb{R}^2$, luego, en condiciones favorables, la convergencia es cuadrática.

Ejemplo 4.10. Aplicar el método de Bairstow para hallar las raíces de

$$p(x) = 4x^5 + 5x^4 + x^3 - x + 2\,,$$

con $d^0 = 2$, $e^0 = -3$ y $\varepsilon = 10^{-8}$.

```
k = 0
             4.0000    5.0000    1.0000    0.0000   -1.0000    2.0000
   2.0000              8.0000   26.0000   30.0000  -18.0000 -128.0000
  -3.0000                       -12.0000  -39.0000  -45.0000   27.0000
         ------------------------------------------------------------
             4.0000   13.0000   15.0000   -9.0000  -64.0000  -99.0000
   2.0000              8.0000   42.0000   90.0000   36.0000
  -3.0000                       -12.0000  -63.0000 -135.0000
         -------------------------------------------------
```

```
              4.0000   21.0000   45.0000   18.0000 -163.0000
J
   18.0000   45.0000
 -163.0000   18.0000
Delta :   -0.4313    1.5947
d, e  :    1.5687   -1.4053
=====================================================================
k = 1
              4.0000    5.0000    1.0000    0.0000   -1.0000    2.0000
    1.5687              6.2750   17.6875   20.4979    7.3000  -18.9220
   -1.4053                       -5.6211  -15.8444  -18.3619   -6.5393
          ---------------------------------------------------------
              4.0000   11.2750   13.0664    4.6534  -12.0619  -23.4613
    1.5687              6.2750   27.5313   54.8694   54.6869
   -1.4053                       -5.6211  -24.6625  -49.1518
          ----------------------------------------------
              4.0000   17.5499   34.9767   34.8603   -6.5268
J
   34.8603   34.9767
   -6.5268   34.8603
Delta :   -0.2772    0.6211
d, e  :    1.2916   -0.7842
=====================================================================
k = 2
              4.0000    5.0000    1.0000    0.0000   -1.0000    2.0000
    1.2916              5.1662   13.1303   14.1990    8.0426   -2.0383
   -0.7842                       -3.1366   -7.9720   -8.6208   -4.8830
          ---------------------------------------------------------
              4.0000   10.1662   10.9937    6.2271   -1.5782   -4.9213
    1.2916              5.1662   19.8029   35.7245   38.6544
   -0.7842                       -3.1366  -12.0231  -21.6898
          ----------------------------------------------
              4.0000   15.3325   27.6599   29.9284   15.3864
J
   29.9284   27.6599
   15.3864   29.9284
Delta :   -0.1891    0.2616
d, e  :    1.1025   -0.5225
=====================================================================
k = 3
              4.0000    5.0000    1.0000    0.0000   -1.0000    2.0000
    1.1025              4.4099   10.3743   10.2357    5.8639    0.0141
   -0.5225                       -2.0901   -4.9168   -4.8511   -2.7792
          ---------------------------------------------------------
              4.0000    9.4099    9.2842    5.3188    0.0128   -0.7651
```

```
    1.1025             4.4099   15.2361   24.7289   25.1660
   -0.5225                      -2.0901   -7.2211  -11.7202
          ---------------------------------------------------
             4.0000   13.8198   22.4303   22.8267   13.4586
J
   22.8267   22.4303
   13.4586   22.8267
Delta :   -0.0796    0.0805
d, e  :    1.0229   -0.4420
=========================================================================
k = 4
             4.0000    5.0000    1.0000    0.0000   -1.0000    2.0000
    1.0229             4.0914    9.2992    8.7259    4.8147    0.0445
   -0.4420                      -1.7682   -4.0189   -3.7711   -2.0808
          ---------------------------------------------------------------
             4.0000    9.0914    8.5310    4.7071    0.0435   -0.0362
    1.0229             4.0914   13.4841   20.7096   20.0369
   -0.4420                      -1.7682   -5.8275   -8.9501
          ---------------------------------------------------
             4.0000   13.1828   20.2469   19.5892   11.1303
J
   19.5892   20.2469
   11.1303   19.5892
Delta :   -0.0100    0.0075
d, e  :    1.0128   -0.4345
=========================================================================
k = 5
             4.0000    5.0000    1.0000    0.0000   -1.0000    2.0000
    1.0128             4.0513    9.1675    8.5377    4.6639    0.0012
   -0.4345                      -1.7380   -3.9329   -3.6627   -2.0008
          ---------------------------------------------------------------
             4.0000    9.0513    8.4295    4.6048    0.0012    0.0004
    1.0128             4.0513   13.2709   20.2186   19.3757
   -0.4345                      -1.7380   -5.6932   -8.6738
          ---------------------------------------------------
             4.0000   13.1027   19.9623   19.1302   10.7032
J
   19.1302   19.9623
   10.7032   19.1302
Delta :   -0.0001    0.0000
d, e  :    1.0127   -0.4345
=========================================================================
k = 6
             4.0000    5.0000    1.0000    0.0000   -1.0000    2.0000
    1.0127             4.0509    9.1662    8.5357    4.6619    0.0000
```

```
-0.4345                      -1.7379   -3.9324   -3.6619   -2.0000
      ------------------------------------------------------------
          4.0000    9.0509    8.4283    4.6033    0.0000    0.0000
```

Entonces

$$\begin{aligned} d &= 1.0127362 \\ e &= -0.4344745 \\ x^2 - 1.0127362\,x + 0.4344745 &\quad \text{divide a } p\,, \\ r_1 &= 0.5063681 + 0.4219784\,i \quad \text{es raíz de } \ p, \\ r_2 &= 0.5063681 - 0.4219784\,i \quad \text{es raíz de } \ p, \\ q(x) &= 4\,x^3 + 9.0509449\,x^2 + 8.4283219\,x + 4.6032625\,. \end{aligned}$$

Al aplicar el método de Bairstow a $q(x)$ con $d^0 = -1$ y $e^0 = -1$ se obtiene:

$$\begin{aligned} d &= -0.9339455 \\ e &= -0.8660624 \\ x^2 + 0.9339455\,x + 0.8660624 &\quad \text{divide a } p\,, \\ r_3 &= -0.4669728 + 0.8049837\,i \quad \text{es raíz de } \ p, \\ r_4 &= -0.4669728 - 0.8049837\,i \quad \text{es raíz de } \ p, \\ \tilde{q}(x) &= 4\,x + 5.3151629\,. \end{aligned}$$

La última raíz es $r_5 = -1.3287907\,.$

Ejercicios

Trate de resolver las ecuaciones propuestas, utilice métodos diferentes, compare sus ventajas y desventajas. Emplee varios puntos iniciales. Busque, si es posible, otras raíces.

4.1 $x^3 + 2x^2 + 3x + 4 = 0.$

4.2 $x^3 + 2x^2 - 3x - 4 = 0.$

4.3 $x^4 - 4x^3 + 6x^2 - 4x + 1 = 0.$

4.4 $x^4 - 4x^3 + 6x^2 - 4x - 1 = 0.$

4.5 $x^4 - 4x^3 + 6x^2 - 4x + 2 = 0.$

4.6

$$\frac{\dfrac{3x-6}{\cos(x)+2} - \dfrac{x-2}{x^2+1}}{\dfrac{x^2+x+10}{e^x+x^2}} + x^3 - 8 = 0.$$

4.7

$$1000000\, i\, \frac{(1+i)^{12}}{(1+i)^{12}-1} = 945560.$$

4.8 $x_1^2 - x_1x_2 + 3x_1 - 4x_2 + 10 = 0,$
$-2x_1^2 + x_2^2 + 3x_1x_2 - 4x_1 + 5x_2 - 42 = 0.$

4.9 $x_1 + x_2 + 2x_1x_2 - 31 = 0,$
$6x_1 + 5x_2 + 3x_1x_2 - 74 = 0.$

5

Interpolación y aproximación

En muchas situaciones de la vida real se tiene una tabla de valores correspondientes a dos magnitudes relacionadas; por ejemplo,

Año	Población
1930	3425
1940	5243
1950	10538
1960	19123
1970	38765
1980	82468
1985	91963
1990	103646
1995	123425

De manera más general, se tiene una tabla de valores

x_1	$f(x_1)$
x_2	$f(x_2)$
$\vdots$	$\vdots$
x_n	$f(x_n)$

y se desea obtener una función $\tilde{f}$, sencilla y fácil de calcular, aproximación de f, o en otros casos, dado un $\bar{x}$, se desea obtener $\tilde{f}(\bar{x})$ valor aproximado de $f(\bar{x})$.

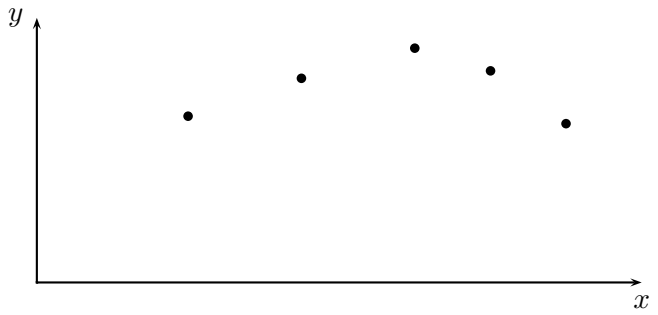


Figura 5.1: Puntos o datos iniciales

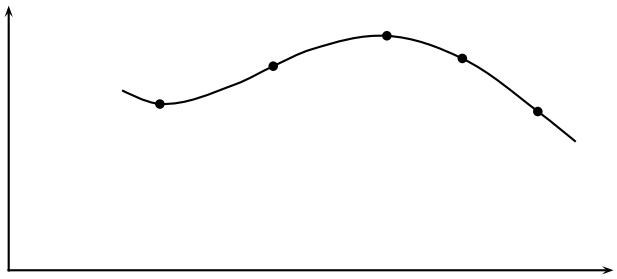


Figura 5.2: Interpolación

Los valores $f(x_i)$ pueden corresponder a:

- Datos o medidas obtenidos experimentalmente.
- Valores de una función f que se conoce pero tiene una expresión analítica muy complicada o de evaluación difícil o lenta.
- Una función de la que no se conoce una expresión analítica, pero se puede conocer $f(x)$ como solución de una ecuación funcional (por ejemplo, una ecuación diferencial) o como resultado de un proceso numérico.

Cuando se desea que la función $\tilde{f}$ pase exactamente por los puntos conocidos,

$$\tilde{f}(x_i) = f(x_i) \, \forall i,$$

se habla de *interpolación* o de *métodos de colocación*, figura 5.2.

En los demás casos se habla de aproximación, figura 5.3. En este capítulo se verá aproximación por mínimos cuadrados.

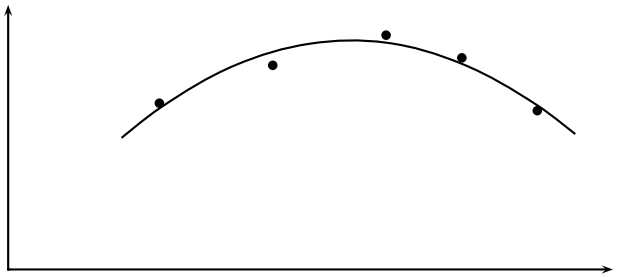


Figura 5.3: Aproximación

5.1 Interpolación

5.1.1 En Scilab

Cuando hay m puntos (x_1, y_1), (x_2, y_2), ..., (x_m, y_m) se desea obtener la función interpolante, una función que pase por esos puntos, con el objetivo de evaluarla en otros valores x intermedios.

La función `interpln` permite hacer interpolación lineal (la función interpolante es continua y afín por trozos). Tiene dos parámetros, el primero es una matriz de dos filas. La primera fila tiene los valores x_i. Deben estar en orden creciente. La segunda fila tiene los valores y_i. El segundo parámetro es un vector donde están los valores x en los que se desea evaluar la función interpolante (afín por trozos).

```
clear, clf
x = [ 0.5 1 1.5 2.1 3 3.6]'
y = [ 1 2 1.5 2.5 2.1 2.4]'

t = 0.8
ft = interpln( [x'; y'], t)

n = length(x);
xx = ( x(1):0.1:x(n) )';
y1 = interpln( [x'; y'], xx);

plot2d(xx, y1)
```

La gráfica resultante es semejante a la de la figura 5.4.

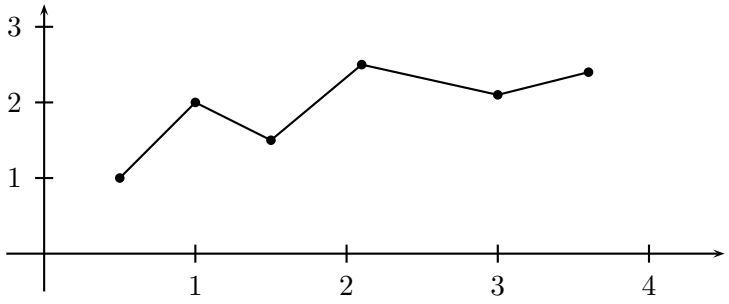


Figura 5.4: Interpolación lineal con `interpln`

También se puede hacer interpolación utilizando funciones *spline* o trazadores cúbicos. Para hacer esto en Scilab, se requieren dos pasos. En el primero, mediante `splin`, a partir de un lista de puntos (x_1, y_1), (x_2, y_2), ..., (x_m, y_m) se calculan las derivadas, en los puntos x_i, de la función *spline* interpolante.

En el segundo paso, mediante `interp`, se evalúa la función interpolante en los valores dados por un vector, primer parámetro de `interp`.

```
clear, clf
x = [ 0.5 1 1.5 2.1 3 3.6]'
y = [ 1 2 1.5 2.5 2.1 2.4]'

n = length(x);
xx = ( x(1):0.1:x(n) )';

d = splin(x, y);
ys = interp(xx, x, y, d);
plot2d(xx, ys)
```

La gráfica resultante es semejante a la de la figura 5.5.

5.1.2 Caso general

En el caso general de interpolación se tiene un conjunto de n puntos (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) con la condición de que los x_i son todos diferentes. Este conjunto se llama el soporte. La función $\tilde{f}$, que se desea construir, debe ser combinación lineal de n funciones llamadas funciones de la base.

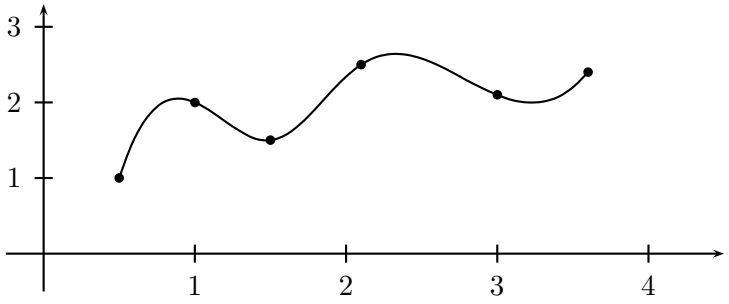


Figura 5.5: Interpolación con funciones *spline*

Supongamos que estas funciones son φ_1, φ_2, ..., φ_n. Entonces,

$$\tilde{f}(x) = a_1\varphi_1(x) + a_2\varphi_2(x) + \cdots + a_n\varphi_n(x).$$

Como las funciones de la base son conocidas, para conocer $\tilde{f}$ basta conocer los escalares a_1, a_2, ..., a_n.

Las **funciones de la base deben ser linealmente independientes**. Si $n \geq 2$, la independencia lineal significa que no es posible que una de las funciones sea combinación lineal de las otras. Por ejemplo, las funciones $\varphi_1(x) = 4$, $\varphi_2(x) = 6x^2 - 20$ y $\varphi_3(x) = 2x^2$ no son linealmente independientes.

Los escalares a_1, a_2, ..., a_n se escogen de tal manera que $\tilde{f}(x_i) = y_i$, para $i = 1, 2, ..., n$. Entonces

$$\begin{aligned} a_1\varphi_1(x_1) + a_2\varphi_2(x_1) + \cdots + a_n\varphi_n(x_1) &= y_1 \\ a_1\varphi_1(x_2) + a_2\varphi_2(x_2) + \cdots + a_n\varphi_n(x_2) &= y_2 \\ &\vdots \\ a_1\varphi_1(x_n) + a_2\varphi_2(x_n) + \cdots + a_n\varphi_n(x_n) &= y_n \end{aligned}$$

Las m igualdades anteriores se pueden escribir matricialmente:

$$\begin{bmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_n(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_n(x_2) \\ \vdots & & & \\ \varphi_1(x_n) & \varphi_2(x_n) & \cdots & \varphi_n(x_n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

De manera compacta se tiene

$$\Phi\, a = y. \tag{5.1}$$

La matriz Φ es una matriz cuadrada $n \times n$, a es un vector columna $n \times 1$, y es un vector columna $n \times 1$. Son conocidos la matriz Φ y el vector columna y. El vector columna a es el vector de incógnitas. Como las funciones de la base son linealmente independientes, entonces las columnas de Φ son linealmente independientes. En consecuencia, Φ es invertible y (5.1) se puede resolver (numéricamente).

Ejemplo 5.1. Dados los puntos $(-1, 1)$, $(2, -2)$, $(3, 5)$ y la base formada por las funciones $\varphi_1(x) = 1$, $\varphi_2(x) = x$, $\varphi_3(x) = x^2$, encontrar la función de interpolación.

Al plantear $\Phi a = y$, se tiene

$$\begin{bmatrix} 1 & -1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix}$$

Entonces

$$a = \begin{bmatrix} -4 \\ -3 \\ 2 \end{bmatrix}, \quad \tilde{f}(x) = -4 - 3x + 2x^2,$$

que efectivamente pasa por los puntos dados. $\diamond$

La *interpolación polinomial* (las funciones utilizadas son 1, x, x^2, ...) para problemas pequeños con matrices "sin problemas", se puede realizar en Scilab, mediante órdenes semejantes a:

```
x = [ 0.5 1 1.5 2.1 3 3.6]'
y = [ 1 2 1.5 2.5 2.1 2.4]'

x = x(:);
y = y(:);
n = size(x,1);
n1 = n - 1;

F = ones(n,n);
for i=1:n1
  F(:,i+1) = x.^i;
end
a = F\y
p = poly(a, 'x', 'c')
```

```
xx = (x(1):0.05:x(n))';
yp = horner(p, xx);
```

Hay ejemplos clásicos de los problemas que se pueden presentar con valores relativamente pequeños, $n = 20$.

Ejemplo 5.2. Dados los puntos mismos $(-1, 1)$, $(2, -2)$, $(3, 5)$ y la base formada por las funciones $\varphi_1(x) = 1$, $\varphi_2(x) = e^x$, $\varphi_3(x) = e^{2x}$, encontrar la función de interpolación.

Al plantear $\Phi a = y$, se tiene

$$\begin{bmatrix} 1 & 0.3679 & 0.1353 \\ 1 & 7.3891 & 54.5982 \\ 1 & 20.0855 & 403.4288 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix}$$

Entonces

$$a = \begin{bmatrix} -1.2921 \\ -0.8123 \\ 0.0496 \end{bmatrix}, \quad \tilde{f}(x) = 1.2921 - 0.8123e^x + 0.0496e^{2x},$$

que efectivamente también pasa por los puntos dados. $\diamond$

5.2 Interpolación polinomial de Lagrange

En la interpolación de Lagrange la función $\tilde{f}$ que pasa por los puntos es un polinomio, pero el polinomio se calcula utilizando polinomios de Lagrange, sin resolver explícitamente un sistema de ecuaciones. Teóricamente, el polinomio obtenido por interpolación polinomial (solución de un sistema de ecuaciones) es exactamente el mismo obtenido por interpolación de Lagrange.

Dados n puntos

$$(x_1, y_1),\ (x_2, y_2),\ \ldots,\ (x_n, y_n),$$

donde $y_i = f(x_i) = f_i$, se desea encontrar un polinomio $p \in \mathcal{P}_{n-1}$ (el conjunto de polinomios de grado menor o igual a $n-1$), que pase exactamente por esos puntos, es decir,

$$p(x_i) = y_i\ ,\ \ i = 1, 2, ..., n. \tag{5.2}$$

Por ejemplo, se desea encontrar un polinomio de grado menor o igual a 2 que pase por los puntos

$$(-1, 1),\ (2, -2),\ (3, 5).$$

Los valores x_i deben ser todos diferentes entre sí. Sin perder generalidad, se puede suponer que $x_1 < x_2 < \cdots < x_n$.

El problema 5.2 se puede resolver planteando n ecuaciones con n incógnitas (los coeficientes del polinomio). Este sistema lineal se puede resolver y se tendría la solución. Una manera más adecuada de encontrar p es por medio de los polinomios de Lagrange.

5.2.1 Algunos resultados previos

Teorema 5.1. *Sea $p \in \mathcal{P}_{n-1}$. Si existen n valores diferentes x_1, x_2, ..., x_n tales que $p(x_i) = 0\ \forall i$, entonces $p(x) = 0\ \forall x$, es decir, p es el polinomio nulo.*

Teorema 5.2. ***Teorema del valor medio**. Sea f derivable en el intervalo $[a, b]$, entonces existe $c \in [a, b]$ tal que*

$$\frac{f(b) - f(a)}{b - a} = f'(c).$$

Corolario 5.1. *Si $f(a) = f(b) = 0$, entonces existe $c \in [a, b]$ tal que*

$$f'(c) = 0.$$

5.2.2 Polinomios de Lagrange

Dados n valores diferentes $x_1, x_2, ..., x_n$, se definen n polinomios de Lagrange $L_1, L_2, ..., L_n$ de la siguiente manera:

$$L_k(x) = \frac{\prod_{i=1, i\neq k}^{n} (x - x_i)}{\prod_{i=1, i\neq k}^{n} (x_k - x_i)}. \tag{5.3}$$

La construcción de los polinomios de Lagrange, para los datos del último ejemplo $x_1 = -1$, $x_2 = 2$, $x_3 = 3$, da:

$$\begin{aligned} L_1(x) &= \frac{(x-2)(x-3)}{(-1-2)(-1-3)} = \frac{x^2-5x+6}{12}, \\ L_2(x) &= \frac{(x--1)(x-3)}{(2--1)(2-3)} = \frac{x^2-2x-3}{-3}, \\ L_3(x) &= \frac{(x--1)(x-2)}{(3--1)(3-2)} = \frac{x^2-x-2}{4}. \end{aligned}$$

Es claro que el numerador de (5.3) es el producto de $n-1$ polinomios de grado 1; entonces el numerador es un polinomio de grado, exactamente, $n-1$. El denominador es el producto de $n-1$ números, ninguno de los cuales es nulo, luego el denominador es un número no nulo. En resumen, L_k es un polinomio de grado $n-1$.

Reemplazando se verifica que L_k se anula en todos los x_i, salvo en x_k,

$$L_k(x_i) = \begin{cases} 0 & \text{si} \quad i \neq k, \\ 1 & \text{si} \quad i = k. \end{cases} \tag{5.4}$$

En el ejemplo, $L_3(-1) = 0$, $L_3(2) = 0$, $L_3(3) = 1$.

Con los polinomios de Lagrange se construye inmediatamente p,

$$p(x) = \sum_{k=1}^{n} y_k L_k(x). \tag{5.5}$$

Por construcción p es un polinomio en $\mathcal{P}_{n-1}$. Reemplazando, fácilmente se verifica 5.2.

Para el ejemplo,

$$p(x) = 1L_1(x) - 2L_2(x) + 5L_3(x) = 2x^2 - 3x - 4.$$

Ejemplo 5.3. Encontrar el polinomio, de grado menor o igual a 3, que pasa por los puntos

$$(-1, 1),\ (1, -5),\ (2, -2),\ (3, 5).$$

$$\begin{aligned} L_1(x) &= \frac{(x-1)(x-2)(x-3)}{(-1-1)(-1-2)(-1-3)} = \frac{x^3-6x^2+11x-6}{-24}, \\ L_2(x) &= \frac{x^3-4x^2+x+6}{4}, \\ L_3(x) &= \frac{x^3-3x^2-x+3}{-3}, \\ L_4(x) &= \frac{x^3-2x^2-x+2}{8}, \\ p(x) &= 2x^2-3x-4. \ \diamond \end{aligned}$$

En la práctica se usa la interpolación de Lagrange de grado 2 o 3, máximo 4. Si hay muchos puntos, éstos se utilizan por grupos de 3 o 4, máximo 5 puntos.

Ejemplo 5.4. Considere los puntos

(1, 3.8), (2, 3.95), (3, 4.), (4, 3.95), (4.2, 3.43), (4.5, 3.89).

El polinomio de interpolación es

$$\begin{aligned} p(x) = &-102.68595 + 245.23493x - 204.16498x^2 + 78.696263x^3 \\ &- 14.264007x^4 + 0.9837509x^5 \end{aligned}$$

Obviamente $p(1) = 3.8$ y $p(2) = 3.95$. Sin embargo $p(1.35) = 6.946$. Ver figura (5.6). $\diamond$

Si `x` es un vector, un polinomio de Lagrange se puede costruir en Scilab por órdenes semejantes a

```
x = [-1 1 2 3]';
n = length(x)
k = 2

Lk = poly([1], 'x', 'c');
deno = 1;
for i=1:n
  if i ~= k
    Lk = Lk*poly([x(i)], 'x');
    deno = deno*(x(k) - x(i));
  end
end
Lk = Lk/deno
```

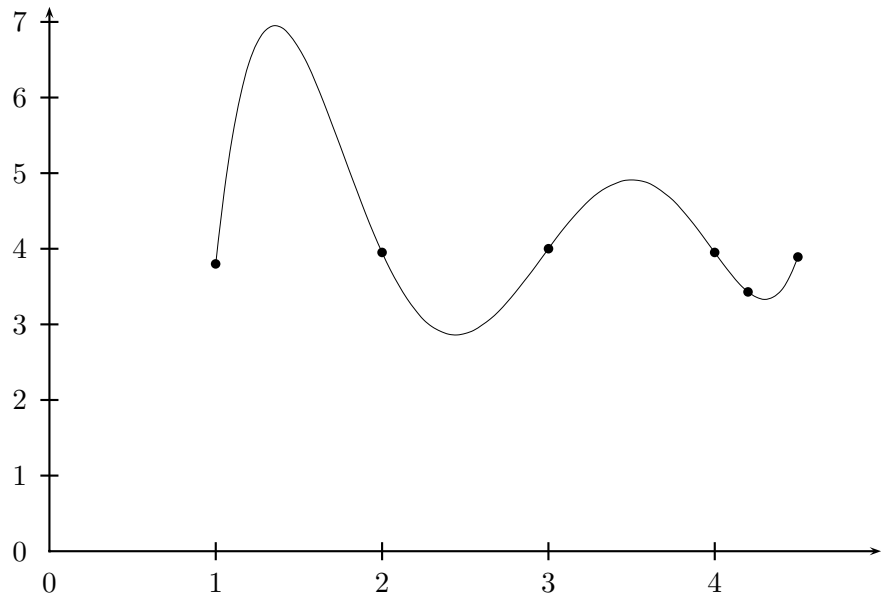


Figura 5.6: Un ejemplo de interpolación polinomial

5.2.3 Existencia, unicidad y error

El polinomio $p \in \mathcal{P}_{n-1}$ existe puesto que se puede construir. Sea $q \in \mathcal{P}_{n-1}$ otro polinomio tal que

$$q(x_i) = y_i \ , \ \ i = 1, 2, ..., n.$$

Sea $r(x) = p(x) - q(x)$. Por construcción, $r \in \mathcal{P}_n$, además $r(x_i) = 0$, $i = 1, 2, n$, o sea, r se anula en n valores diferentes, luego $r(x) = 0$, de donde $q(x) = p(x)$.

Teorema 5.3. *Sean x_1, x_2, ..., x_n reales distintos; t un real; I_t el menor intervalo que contiene a x_1, x_2, ..., x_n, t; $f \in C^n_{I_t}$ (f tiene derivadas continuas de orden 0, 1, 2, ..., n); p_{n-1} el polinomio de grado menor o igual a $n-1$ que pasa por los n puntos $(x_1, f(x_1))$, ..., $(x_n, f(x_n))$. Entonces $E(t)$, el error en t, está dado por:*

$$E(t) = f(t) - p_{n-1}(t) = (t - x_1)(t - x_2) \cdots (t - x_n) f^{(n)}(\xi)/n! \qquad (5.6)$$

para algún $\xi \in I_t$.

Demostración. Si $t = x_i$ para algún i, entonces se tiene trivialmente el resultado. Supongamos ahora que $t \notin \{x_1, x_2, ..., x_n\}$. Sean

$$\begin{aligned} \Phi(x) &= (x - x_1)(x - x_2) \cdots (x - x_n), \\ G(x) &= E(x) - \frac{\Phi(x)}{\Phi(t)} E(t). \end{aligned}$$

Entonces

$$\begin{aligned} G &\in C^n_{I_t}, \\ G(x_i) &= E(x_i) - \frac{\Phi(x_i)}{\Phi(t)} E(t) = 0, \quad i = 1, ..., n \\ G(t) &= E(t) - \frac{\Phi(t)}{\Phi(t)} E(t) = 0. \end{aligned}$$

Como G tiene por lo menos $n + 1$ ceros en I_t, aplicando el corolario del teorema del valor medio, se deduce que G' tiene por lo menos $n+1-1$ ceros en I_t. Así sucesivamente se concluye que $G^{(n)}$ tiene por lo menos un cero en I_t. Sea ξ tal que

$$G^{(n)}(\xi) = 0.$$

De acuerdo con las definiciones

$$\begin{aligned} E^{(n)}(x) &= f^{(n)}(x) - p_n^{(n)}(x) = f^{(n)}(x), \\ \Phi^{(n)}(x) &= n!, \\ G^{(n)}(x) &= E^{(n)}(x) - \frac{\Phi^{(n)}(x)}{\Phi(t)} E(t), \\ G^{(n)}(x) &= f^{(n)}(x) - \frac{n!}{\Phi(t)} E(t), \\ G^{(n)}(\xi) &= f^{(n)}(\xi) - \frac{n!}{\Phi(t)} E(t) = 0. \end{aligned}$$

Entonces

$$E(t) = \frac{\Phi(t)}{n!} f^{(n)}(\xi). \quad \square$$

Frecuentemente no se tiene la información necesaria para aplicar (5.6). Algunas veces se tiene información necesaria para obtener una cota superior del valor absoluto del error.

$$|E(t)| \leq \frac{|\Phi(t)|}{n!} \max_{z \in I_t} |f^{(n)}(z)| \tag{5.7}$$

Ejemplo 5.5. Considere los valores de la función seno en los puntos 5, 5.2, 5.5 y 6. Sea p el polinomio de interpolación. Obtenga una cota para error cometido al aproximar sen(5.8) por $p(5.8)$. Compare con el valor real del error.

$$y = (-0.9589243,\ -0.8834547,\ -0.7055403,\ -0.2794155).$$

El polinomio p se puede obtener mediante la solución de un sistema de ecuaciones o por polinomios de Lagrange.

$$\begin{aligned} p(x) &= 23.728487 - 12.840218\,x + 2.117532\,x^2 - 0.1073970\,x^3 \\ p(5.8) &= -0.4654393 \\ f^{(4)}(x) &= \operatorname{sen}(x) \\ I_t &= [5, 6] \\ \max_{z \in I_t} |f^{(n)}(z)| &= 0.9589243 \\ |\Phi(5.8)| &= 0.0288 \\ |E(5.8)| &\leq 0.0011507 \end{aligned}$$

El error cometido es:

$$E(5.8) = \operatorname{sen}(5.8) - p(5.8) = 0.0008371\,.\ \diamond$$

5.3 Diferencias divididas de Newton

Esta es una manera diferente de hacer los cálculos para la interpolación polinómica. En la interpolación de Lagrange se construye explícitamente p, es decir, se conocen sus coeficientes. Por medio de las diferencias divididas no se tiene explícitamente el polinomio, pero se puede obtener fácilmente el valor $p(x)$ para cualquier x.

Supongamos de nuevo que tenemos los mismos n puntos,

$$(x_1, f_1),\ (x_2, f_2),\ \ldots,\ (x_{n-1}, f_{n-1}),\ (x_n, f_n).$$

Con ellos se obtiene $p = p_{n-1} \in \mathcal{P}_{n-1}$. Si se consideran únicamente los primeros $n-1$ puntos

$$(x_1, f_1),\ (x_2, f_2),\ \ldots,\ (x_{n-1}, f_{n-1}),$$

se puede construir $p_{n-2} \in \mathcal{P}_{n-2}$. Sea $c(x)$ la corrección que permite pasar de p_{n-2} a p_{n-1},

$$p_{n-1}(x) = p_{n-2}(x) + c(x), \quad \text{es decir,} \quad c(x) = p_{n-1}(x) - p_{n-2}(x).$$

Por construcción, c es un polinomio en $\mathcal{P}_{n-1}$. Además,

$$c(x_i) = p_{n-1}(x_i) - p_{n-2}(x_i) = 0, \;\; i = 1, 2, ..., n-1.$$

La fórmula anterior dice que c tiene $n-1$ raíces diferentes $x_1, x_2, ..., x_{n-1}$, entonces

$$c(x) = \alpha_{n-1}(x - x_1)(x - x_2)\cdots(x - x_{n-1}).$$

$$\begin{aligned} f(x_n) &= p_{n-1}(x_n) = p_{n-2}(x_n) + c(x_n), \\ f(x_n) &= p_{n-2}(x_n) + \alpha_{n-1}(x_n - x_1)(x_n - x_2)(x_n - x_3)\cdots(x_n - x_{n-1}). \end{aligned}$$

De la última igualdad se puede despejar α_{n-1}. Este valor se define como la diferencia dividida de orden $n-1$ de f en los puntos x_1, x_2, ..., x_n. Se denota

$$\alpha_{n-1} = \; f[x_1, x_2, ..., x_n] := \frac{f(x_n) - p_{n-2}(x_n)}{(x_n - x_1)(x_n - x_2)\cdots(x_n - x_{n-1})}.$$

El nombre diferencia dividida no tiene, por el momento, un significado muy claro; éste se verá más adelante. Una de las igualdades anteriores se reescribe

$$p_{n-1}(x) = p_{n-2}(x) + f[x_1, ..., x_n](x - x_1)\cdots(x - x_{n-1}). \tag{5.8}$$

Esta fórmula es la que se utiliza para calcular $p_{n-1}(x)$, una vez que se sepa calcular, de manera sencilla, $f[x_1, x_2, ..., x_n]$.

- Para calcular $p(x)$, se empieza calculando $p_0(x)$.
- A partir de $p_0(x)$, con el valor $f[x_1, x_2]$, se calcula $p_1(x)$.
- A partir de $p_1(x)$, con el valor $f[x_1, x_2, x_3]$, se calcula $p_2(x)$.
- A partir de $p_2(x)$, con el valor $f[x_1, x_2, x_3, x_4]$, se calcula $p_3(x)$.
- $\vdots$
- A partir de $p_{n-2}(x)$, con el valor $f[x_1, x_2, ..., x_n]$, se calcula $p_{n-1}(x)$.

Obviamente

$$p_0(x) = f(x_0). \tag{5.9}$$

Por definición, consistente con lo visto antes,

$$f[x_0] := f(x_0),$$

que se generaliza a

$$f[x_i] := f(x_i), \ \ \forall i. \tag{5.10}$$

Las demás diferencias divididas se deducen de (5.8),

$$\begin{aligned} p_1(x) &= p_0(x) + f[x_1, x_2](x - x_1), \\ f[x_1, x_2] &= \frac{p_1(x) - p_o(x)}{x - x_1}. \end{aligned}$$

Para $x = x_2$,

$$\begin{aligned} f[x_1, x_2] &= \frac{p_1(x_2) - p_o(x_2)}{x_2 - x_1}, \\ f[x_1, x_2] &= \frac{f(x_2) - f(x_2)}{x_2 - x_1}, \\ f[x_1, x_2] &= \frac{f[x_2] - f[x_1]}{x_2 - x_1}. \end{aligned}$$

La anterior igualdad se generaliza a

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}. \tag{5.11}$$

Deducción de $f[x_1, x_2, x_3]$:

$$\begin{aligned} p_2(x) &= p_1(x) + f[x_1, x_2, x_3](x - x_1)(x - x_2), \\ f[x_1, x_2, x_3] &= \frac{p_2(x) - p_1(x)}{(x - x_1)(x - x_2)}, \\ x &= x_3, \\ f[x_1, x_2, x_3] &= \frac{p_2(x_3) - p_1(x_3)}{(x_3 - x_1)(x_3 - x_2)}, \\ &= \ldots \\ f[x_1, x_2, x_3] &= \frac{f_1(x_3 - x_2) - f_2(x_3 - x_1) + f_3(x_2 - x_1)}{(x_3 - x_2)(x_3 - x_1)(x_2 - x_1)}. \end{aligned}$$

Por otro lado,

$$\begin{aligned}\frac{f[x_2,x_1]-f[x_1,x_2]}{x_3-x_1} &= \frac{\dfrac{f_3-f_2}{x_3-x_2}-\dfrac{f_2-f_1}{x_2-x_1}}{x_3-x_1},\\ &= \dots\\ \frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1} &= \frac{f_1(x_3-x_2)-f_2(x_3-x_1)+f_3(x_2-x_1)}{(x_3-x_2)(x_3-x_1)(x_2-x_1)}.\end{aligned}$$

Luego

$$f[x_1,x_2,x_3]=\frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}.$$

Generalizando,

$$f[x_i,x_{i+1},x_{i+2}]=\frac{f[x_{i+1},x_{i+2}]-f[x_i,x_{i+1}]}{x_{i+2}-x_i}. \tag{5.12}$$

La generalización para diferencias divididas de orden j es:

$$f[x_i,x_{i+1},...,x_{i+j}]=\frac{f[x_{i+1},...,x_{i+j}]-f[x_i,...,x_{i+j-1}]}{x_{i+j}-x_i}. \tag{5.13}$$

Las fórmulas anteriores dan sentido al nombre diferencias divididas. Cuando no se preste a confusión, se puede utilizar la siguiente notación:

$$D^jf[x_i]:=f[x_i,x_{i+1},...,x_{i+j}]. \tag{5.14}$$

Entonces

$$D^0f[x_i] := f(x_i), \tag{5.15}$$

$$Df[x_i]=D^1f[x_i] = \frac{D^0f[x_{i+1}]-D^0f[x_i]}{x_{i+1}-x_i}, \tag{5.16}$$

$$D^2f[x_i] = \frac{D^1f[x_{i+1}]-D^1f[x_i]}{x_{i+2}-x_i}, \tag{5.17}$$

$$D^jf[x_i] = \frac{D^{j-1}f[x_{i+1}]-D^{j-1}f[x_i]}{x_{i+j}-x_i}. \tag{5.18}$$

5.3.1 Tabla de diferencias divididas

Para ejemplos pequeños, hechos a mano, se acostumbra construir la tabla de diferencias divididas, la cual tiene el siguiente aspecto:

x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
x_1	f_1			
		$f[x_1, x_2]$		
x_2	f_2		$f[x_1, x_2, x_3]$	
		$f[x_2, x_3]$		$f[x_1, x_2, x_3, x_4]$
x_3	f_3		$f[x_2, x_3, x_4]$	
		$f[x_3, x_4]$		$f[x_2, x_3, x_4, x_5]$
x_4	f_4		$f[x_3, x_4, x_5]$	
		$f[x_4, x_5]$		
x_5	f_5			

En la tabla anterior, dados 5 puntos, están las diferencias divididas hasta de orden 3. Claro está, se hubiera podido calcular también la diferencia dividida de orden 4, que estaría colocada en una columna adicional a la derecha.

La elaboración de la tabla es relativamente sencilla. Las dos primeras columnas corresponden a los datos. La tercera columna, la de las diferencias divididas de primer orden, $f[x_i, x_{i+1}]$, se obtiene mediante la resta de dos elementos consecutivos de la columna anterior dividida por la resta de los dos elementos correspondientes de la primera columna. Por ejemplo, $f[x_3, x_4] = (f_4 - f_3)/(x_4 - x_3)$. Obsérvese que este valor se coloca en medio de la fila de f_3 y de la fila de f_4.

Para el cálculo de una diferencia dividida de segundo orden, cuarta columna, se divide la resta de dos elementos consecutivos de la columna anterior por la resta de dos elementos de la primera columna, pero dejando uno intercalado. Por ejemplo, $f[x_1, x_2, x_3] = (f[x_2, x_3] - f[x_1, x_2])/(x_3 - x_1)$.

Para el cálculo de una diferencia dividida de tercer orden, quinta columna, se divide la resta de dos elementos consecutivos de la columna anterior por la resta de dos elementos de la primera columna, pero dejando dos intercalados. Por ejemplo, $f[x_1, x_2, x_3, x_4] = (f[x_2, x_3, x_4] - f[x_1, x_2, x_3])/(x_4 - x_1)$.

Ejemplo 5.6. Construir la tabla de diferencias divididas, hasta el orden 3, a partir de los seis puntos siguientes:

$$(0,\ 0),\ (0.5,\ 0.7071),\ (1,\ 1),\ (2,\ 1.4142),\ (3,\ 1.7321),\ (4,\ 2).$$

x_i	f_i	$Df[x_i]$	$D^2f[x_i]$	$D^3f[x_i]$
0	0.0000			
		1.4142		
.5	0.7071		-0.8284	
		0.5858		0.3570
1	1.0000		-0.1144	
		0.4142		0.0265
2	1.4142		-0.0482	
		0.3179		0.0077
3	1.7321		-0.0250	
		0.2679		
4	2.0000			

El valor 1.4142 es simplemente $(0.7071 - 0)/(0.5 - 0)$. El valor 0.2679 es simplemente $(2-1.7321)/(4-3)$. El valor -0.1144 es simplemente $(0.4142-.5858)/(2-.5)$. El valor 0.0077 es simplemente $(-0.0250 - -0.0482)/(4-1)$. $\diamond$

El esquema algorítmico para calcular la tabla de diferencias divididas hasta el orden m es el siguiente:

para $i = 1, ..., n$
    $D^0 f[x_i] = f(x_i)$
fin-para i
para $j = 1, ..., m$
    para $i = 1, ..., n - j$
        calcular $D^j f[x_i]$ según (5.18)
    fin-para i
fin-para j

Suponiendo que `x`, `y` son vectores y que se conoce `m`, la tabla de diferencias divididas, hasta el orden `m`, se puede costruir en Scilab por órdenes semejantes a:

```
x = x(:)
y = y(:)
n = size(x,1)

DD = zeros(n,m+1);
```

```
DD(:,1) = y;
for j=1:m
  for i=1:n-j
    Djfi = ( DD(i+1,j) - DD(i,j) )/( x(i+j) - x(i) );
    DD(i,j+1) = Djfi;
  end
end
disp(DD)
```

Si los datos $f(x_i)$ corresponden a un polinomio, esto se puede deducir mediante las siguientes observaciones:

- Si para algún m todos los valores $f[x_k, x_{k+1}, ..., x_{k+m}]$ son iguales (o aproximadamente iguales), entonces f es (aproximadamente) un polinomio de grado m.
- Si para algún r todos los valores $f[x_k, x_{k+1}, ..., x_{k+r}]$ son nulos (o aproximadamente nulos), entonces f es (aproximadamente) un polinomio de grado $r-1$.

5.3.2 Cálculo del valor interpolado

La fórmula (5.8) se puede reescribir a partir de un punto x_k, pues no siempre se debe tomar como valor de referencia x_1,

$$p_m(x) = p_{m-1}(x) + D^m f[x_k](x - x_k)(x - x_{k+1}) \cdots (x - x_{k+m-1}). \quad (5.19)$$

Si se calcula $p_{m-1}(x)$ de manera análoga, queda en función de $p_{m-2}(x)$ y así sucesivamente se obtiene:

$$p_m(x) = \sum_{i=0}^{m} \left[D^i f[x_k] \prod_{j=0}^{i-1} (x - x_{k+j}) \right]. \quad (5.20)$$

El proceso para el cálculo es el siguiente:

$$\begin{aligned}
&p_0(x) = f_k \\
&p_1(x) = p_0(x) + D^1 f[x_k](x - x_k) \\
&p_2(x) = p_1(x) + D^2 f[x_k](x - x_k)(x - x_{k+1}) \\
&p_3(x) = p_2(x) + D^3 f[x_k](x - x_k)(x - x_{k+1})(x - x_{k+2}) \\
&p_4(x) = p_3(x) + D^4 f[x_k](x - x_k)(x - x_{k+1})(x - x_{k+2})(x - x_{k+3}) \\
&\vdots
\end{aligned}$$

Se observa que para calcular $p_j(x)$ hay multiplicaciones que ya se hicieron para obtener $p_{j-1}(x)$; entonces, no es necesario repetirlas sino organizar el proceso de manera más eficiente.

$$\begin{array}{ll}
\gamma_0 = 1, & p_0(x) = f_k \\
\gamma_1 = \gamma_0(x - x_k), & p_1(x) = p_0(x) + D^1 f[x_k]\,\gamma_1 \\
\gamma_2 = \gamma_1(x - x_{k+1}), & p_2(x) = p_1(x) + D^2 f[x_k]\,\gamma_2 \\
\gamma_3 = \gamma_2(x - x_{k+2}), & p_3(x) = p_2(x) + D^3 f[x_k]\,\gamma_3 \\
\gamma_4 = \gamma_3(x - x_{k+3}), & p_4(x) = p_3(x) + D^4 f[x_k]\,\gamma_4 \\
\vdots &
\end{array}$$

Únicamente queda por precisar la escogencia del punto inicial o de referencia x_k. Si se desea evaluar $p_m(\bar{x})$, ¿cuál debe ser x_k? Recordemos que se supone que los puntos x_1, x_2, ..., x_n están ordenados y que m, orden del polinomio de interpolación, es menor o igual que $n-1$. Obviamente, aunque no es absolutamente indispensable, también se supone que $\bar{x} \notin \{x_1, x_2, ..., x_n\}$.

Naturalmente se desea que $\bar{x} \in [x_k, x_{k+m}]$. Pero no siempre se cumple; esto sucede cuando $\bar{x} \notin [x_1, x_n]$. En estos casos se habla de **extrapolación** y se debe escoger $x_k = x_1$ si $\bar{x} < x_1$. En el caso opuesto se toma $x_k = x_{n-m}$.

En los demás casos, se desea que $\bar{x}$ esté lo "más cerca" posible del intervalo $[x_k, x_{k+m}]$ o del conjunto de puntos x_k, x_{k+1}, x_{k+2}, ..., x_{k+m}.

Ejemplo 5.7. Considere los datos del ejemplo anterior para calcular por interpolación cuadrática y por interpolación cúbica una aproximación de $f(1.69)$.

El primer paso consiste en determinar el x_k. Para ello únicamente se tienen en cuenta los valores x_i.

x_i
0
.5
1
2
3
4

Para el caso de la interpolación cuadrática, una simple inspección visual determina que hay dos posibilidades para x_k. La primera es $x_k = 0.5$, intervalo $[0.5, 2]$. La segunda es $x_k = 1$, intervalo $[1, 3]$. ¿Cuál es mejor?

Para medir la cercanía se puede usar la distancia de $\bar{x}$ al promedio de los extremos del intervalo $(x_i + x_{i+2})/2$ (el centro del intervalo) o la distancia de $\bar{x}$ al promedio de todos los puntos $(x_i + x_{i+1} + x_{i+2})/3$. En general

$$u_i = \frac{x_i + x_{i+m}}{2}, \tag{5.21}$$

$$v_i = \frac{x_i + x_{i+1} + x_{i+2} + \cdots + x_{i+m}}{m+1}, \tag{5.22}$$

$$|\bar{x} - u_k| = \min_i\{|\bar{x} - u_i| : \bar{x} \in [x_i, x_{i+m}]\}, \tag{5.23}$$

$$|\bar{x} - v_k| = \min_i\{|\bar{x} - v_i| : \bar{x} \in [x_i, x_{i+m}]\}. \tag{5.24}$$

Los valores u_i y v_i son, de alguna forma, indicadores del centro de masa del intervalo $[x_i, x_{i+m}]$. Con frecuencia, los dos criterios, (5.23) y (5.24), definen el mismo x_k, pero en algunos casos no es así. De todas formas son criterios razonables y para trabajar se escoge un solo criterio, lo cual da buenos resultados. Se puede preferir la utilización de v_i que, aunque requiere más operaciones, tiene en cuenta todos los x_j pertenecientes a $[x_i, x_{i+m}]$.

Los resultados numéricos para la interpolación cuadrática dan:

x_i	u_i	$\lvert\bar{x} - u_i\rvert$	v_i	$\lvert\bar{x} - v_i\rvert$
0				
.5	1.25	0.44	1.1667	0.5233
1	2.00	0.31$\surd$	2.0000	0.3100$\surd$
2				
3				
4				

Para la interpolación cúbica hay tres posibilidades para x_k: 0 , 0.5 y 1.

x_i	u_i	$\lvert\bar{x} - u_i\rvert$	v_i	$\lvert\bar{x} - v_i\rvert$
0	1.00	0.69	0.875	0.815
.5	1.75	0.06$\surd$	1.625	0.065$\surd$
1	2.50	0.81	2.500	0.810
2				
3				
4				

Una vez escogido $x_k = 1$ para obtener la aproximación cuadrática de $f(1.69)$, los cálculos dan:

$$\begin{array}{ll} \gamma_0 = 1, & p_0(x) = 1, \\ \gamma_1 = 1(1.69-1) = 0.69, & p_1(x) = 1 + 0.4142(0.69) = 1.285798 \\ \gamma_2 = 0.69(1.69-2) = -0.2139, & p_2(x) = 1.285798 - 0.0482(-0.2139) \\ & p_2(x) = 1.296097 \end{array}$$

Para la interpolación cúbica, $x_k = 0.5$:

$$\begin{array}{ll} \gamma_0 = 1, & p_0(x) = 0.7071, \\ \gamma_1 = 1(1.69-0.5) = 1.19, & p_1(x) = 0.7071 + 0.5858(1.19) \\ & p_1(x) = 1.404202 \\ \gamma_2 = 1.19(1.69-1) = 0.8211, & p_2(x) = 1.404202 - 0.1144(0.8211) \\ & p_2(x) = 1.310268 \\ \gamma_3 = 0.8211(1.69-2) = -0.254541, & p_3(x) = 1.310268 + 0.0265(-0.254541) \\ & p_3(x) = 1.303523. \ \diamond \end{array}$$

El esquema del algoritmo para calcular $p_m(\bar{x})$, a partir de la tabla de diferencia divididas, es el siguiente:

determinar k
$\texttt{px} = f(x_k)$
$\texttt{gi} = 1.0$
para $j = 1, ..., m$
    $\texttt{gi} = \texttt{gi} * (\bar{x} - x_{k+j-1})$
    $\texttt{px} = \texttt{px} + \texttt{gi} * D^j f[x_k]$
fin-para j

Si `x` es un vector ordenado de manera creciente, `m` el grado del polinomio interpolante y `t` el valor en el que se desea interpolar, el índice k se puede obtener en Scilab por órdenes semejantes a:

```
n = length(x);

if t <= x(1)
  k = 1
else if t >= x(n)
    k = n-m;
  else
```

```
    distmin = 1.0e10;
    k = -1;
    for i=1:n-m
      if ( x(i) <= t & t <= x(i+m) ) | m == 0
        vi = mean(x(i:i+m));
        di = abs( t - vi );
        if di < distmin
           distmin = di;
           k = i;
        end
      end // if
    end // for i
  end // else
end // else
```

Dados los vectores `x` (ordenado) y `y` , el valor `m` (grado del polinomio), si ya se construyó la tabla de diferencias divididas `DD` y se conoce `k`, entonces el valor $p(t)$ se puede calcular en Scilab así:

```
pt = DD(k,1)
gi = 1
for j=1:m
    gi = gi*(t-x(k+j-1))
    pt = pt + gi*DD(k,j+1)
end
```

La escogencia del "mejor" x_k para calcular $p_m(\bar{x})$, con $m < n-1$, es útil cuando se va a evaluar una aproximación de f en pocos puntos, suficientemente separados entre sí. Cuando hay muchos valores $\bar{x}$ para obtener una aproximación de f, puede suceder que dos de los $\bar{x}$ sean cercanos pero al obtener el "mejor" x_k resulten dos x_k diferentes con dos aproximaciones bastante diferentes, cuando se esperaban dos aproximaciones parecidas. En la sección de *splines* hay un ejemplo detallado.

5.4 Diferencias finitas

Cuando los puntos $(x_1, f(x_1))$, $(x_2, f(x_2))$, $(x_3, f(x_3))$, ..., $(x_n, f(x_n))$, están igualmente espaciados en x, es decir, existe un $h > 0$ tal que

$$x_i = x_{i-1} + h,\ i = 2, ..., n$$
$$x_i = x_1 + (i-1)h,\ i = 1, ..., n$$

entonces se pueden utilizar las diferencias finitas, definidas por

$$\Delta^0 f_i = f_i \tag{5.25}$$
$$\Delta f_i = f_{i+1} - f_i \tag{5.26}$$
$$\Delta^{k+1} f_i = \Delta^k(\Delta f_i) = \Delta^k f_{i+1} - \Delta^k f_i \tag{5.27}$$

Algunas de las propiedades interesantes de las diferencias finitas son:

$$\Delta^k f_i = \sum_{j=0}^{k} (-1)^j \binom{k}{j} f_{i+k-j}\,, \tag{5.28}$$
$$f_{i+k} = \sum_{j=0}^{k} \binom{k}{j} \Delta^j f_i\,. \tag{5.29}$$

Las demostraciones se pueden hacer por inducción. La primera igualdad permite calcular $\Delta^k f_i$ sin tener explícitamente los valores $\Delta^{k-1} f_j$. La segunda igualdad permite el proceso inverso al cálculo de las diferencias finitas (se obtienen a partir de los valores iniciales f_p), es decir, obtener un valor f_m a partir de las diferencias finitas.

Para valores igualmente espaciados, las diferencias finitas y las divididas están estrechamente relacionadas.

$$D^0 f[x_i] = f[x_i] = f_i = \Delta^0 f_i$$
$$D^1 f[x_i] = f[x_i, x_{i+1}] = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} = \frac{\Delta^1 f_i}{h}$$
$$D^2 f[x_i] = f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i} = \cdots = \frac{\Delta^2 f_i}{2h^2}$$
$$D^m f[x_i] = f[x_i, ..., x_{i+m}] = \frac{\Delta^m f_i}{m!\, h^m} \tag{5.30}$$

5.4.1 Tabla de diferencias finitas

La tabla de diferencias finitas tiene una estructura análoga a la tabla de diferencias divididas. Se usa para ejemplos pequeños hechos a mano.

x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$
x_1	f_1			
		Δf_1		
x_2	f_2		$\Delta^2 f_1$	
		Δf_2		$\Delta^3 f_1$
x_3	f_3		$\Delta^2 f_2$	
		Δf_3		$\Delta^3 f_2$
x_4	f_4		$\Delta^2 f_3$	
		Δf_4		
x_5	f_5			

La elaboración de la tabla es muy sencilla. Las dos primeras columnas corresponden a los datos. A partir de la tercera columna, para calcular cada elemento se hace la resta de dos elementos consecutivos de la columna anterior. Por ejemplo, $\Delta f_3 = f_4 - f_3$. Obsérvese que este valor se coloca en medio de la fila de f_3 y de la fila de f_4. Por ejemplo, $\Delta^2 f_1 = \Delta f_2 - \Delta f_1$. De manera semejante, $\Delta^3 f_2 = \Delta^2 f_3 - \Delta^2 f_2$.

Ejemplo 5.8. Construir la tabla de diferencias finitas, hasta el orden 3, a partir de los seis puntos siguientes: $(0,\ 0)$, $(0.5,\ 0.7071)$, $(1,\ 1)$, $(1.5,\ 1.2247)$, $(2,\ 1.4142)$, $(2.5,\ 1.5811)$.

x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$
0	0.0000			
		0.7071		
.5	0.7071		−0.4142	
		0.2929		0.3460
1	1.0000		−0.0682	
		0.2247		0.0330
1.5	1.2247		−0.0352	
		0.1895		0.0126
2	1.4142		−0.0226	
		0.1669		
2.5	1.5811			

El valor 0.1895 es simplemente $1.4142 - 1.2247$. El valor 0.0330 es simplemente $-0.0352 - -0.0682$. $\diamond$

El esquema algorítmico para calcular la tabla de diferencias finitas hasta el orden m es el siguiente:

para $i = 1, ..., n$
    $\Delta^0 f_i = f(x_i)$
fin-para i
para $j = 1, ..., m$
    para $i = 1, ..., n - j$
        $\Delta^j f_i = \Delta^{j-1} f_{i+1} - \Delta^{j-1} f_i$
    fin-para i
fin-para j

5.4.2 Cálculo del valor interpolado

Teniendo en cuenta la relación entre diferencias divididas y finitas (5.30), la igualdad (5.20) se puede escribir

$$p_m(x) = \sum_{i=0}^{m} \left[\frac{\Delta^i f_k}{i!\, h^i} \prod_{j=0}^{i-1} (x - x_{k+j}) \right].$$

El valor $i!$ se puede escribir $\prod_{j=0}^{i-1}(j+1)$. Además, sea $s = (x - x_k)/h$, es decir, $x = x_k + sh$. Entonces, $x - x_{k+j} = x_k + sh - x_k - jh = (s - j)h$.

$$\begin{aligned} p_m(x) &= \sum_{i=0}^{m} \left[\frac{\Delta^i f_k}{i!\, h^i} \prod_{j=0}^{i-1} (s - j)h \right] \\ &= \sum_{i=0}^{m} \left[\frac{\Delta^i f_k}{i!} \prod_{j=0}^{i-1} (s - j) \right] \\ &= \sum_{i=0}^{m} \Delta^i f_k \prod_{j=0}^{i-1} \frac{s - j}{j + 1} \end{aligned}$$

Si a y b son enteros no negativos, $a \geq b$, el coeficiente binomial está definido por

$$\binom{a}{b} = \frac{a!}{(a - b)!\, b!}.$$

Desarrollando los factoriales y simplificando se tiene

$$\binom{a}{b} = \frac{a(a-1)(a-2)\cdots(a-b+1)}{1\times 2\times 3\times\cdots\times b} = \frac{a(a-1)(a-2)\cdots(a-b+1)}{b!}$$

Esta última expresión sirve para cualquier valor real a y cualquier entero no negativo b, con la convención de que $\binom{a}{0} = 1$. Entonces,

$$\prod_{j=0}^{i-1} \frac{s-j}{j+1}$$

se puede denotar simplemente por $\binom{s}{i}$ y así

$$p_m(x) = \sum_{i=0}^{m} \Delta^i f_k \binom{s}{i}. \tag{5.31}$$

Este coeficiente $\binom{s}{i}$ guarda propiedades semejantes a las del coeficiente binomial, en particular

$$\binom{s}{i} = \binom{s}{i-1}\frac{s-i+1}{i}.$$

Esto permite su cálculo de manera recurrente

$$\begin{aligned}
\binom{s}{0} &= 1,\\
\binom{s}{1} &= \binom{s}{0}s\\
\binom{s}{2} &= \binom{s}{1}\frac{s-1}{2}\\
\binom{s}{3} &= \binom{s}{2}\frac{s-2}{3}\\
\binom{s}{4} &= \binom{s}{3}\frac{s-3}{4}\\
&\vdots
\end{aligned}$$

Escoger el x_k para interpolar por un polinomio de grado m, se hace como en las diferencias divididas. Como los valores x_i están igualmente espaciados

los valores, u_i y v_i coinciden.

$$\begin{aligned} u_i &= \frac{x_i + x_{i+m}}{2}, \quad i = 1, ..., n-m, \\ |x - u_k| &= \min\{|x - u_i| : i = 1, ..., n-m\}. \end{aligned}$$

Definido el x_k, es necesario calcular s:

$$s = \frac{x - x_k}{h}.$$

El esquema de los cálculos es:

$$\begin{array}{ll} \gamma_0 = 1, & p_0(x) = f_k \\ \gamma_1 = \gamma_0\, s, & p_1(x) = p_0(x) + \Delta^1 f_k\, \gamma_1 \\ \gamma_2 = \gamma_1(s-1)/2, & p_2(x) = p_1(x) + \Delta^2 f_k\, \gamma_2 \\ \gamma_3 = \gamma_2(s-2)/3, & p_3(x) = p_2(x) + \Delta^3 f_k\, \gamma_3 \\ \gamma_4 = \gamma_3(s-3)/4, & p_4(x) = p_3(x) + \Delta^4 f_k\, \gamma_4 \\ \vdots & \end{array}$$

Ejemplo 5.9. Calcular $p_3(1.96)$ y $p_2(1.96)$ a partir de los puntos (0, 0), (0.5, 0.7071), (1, 1), (1.5, 1.2247), (2, 1.4142), (2.5, 1.5811).

La tabla de diferencias finitas es la misma del ejemplo anterior. Para calcular $p_3(1.96)$ se tiene $x_k = x_2 = 1$. Entonces $s = (1.96 - 1)/0.5 = 1.92$.

$$\begin{array}{ll} \gamma_0 = 1, & p_0(x) = f_2 = 1 \\ \gamma_1 = 1(1.92) = 1.92, & p_1(x) = 1 + .2247(1.92) = 1.431424 \\ \gamma_2 = 1.92(1.92-1)/2 = .8832, & p_2(x) = 1.431424 - .0352(.8832) \\ & p_2(x) = 1.400335 \\ \gamma_3 = \gamma_2(1.92-2)/3 = -.023552, & p_3(x) = 1.400335 + .0126(-.023552) \\ & p_3(x) = 1.400039 \end{array}$$

Para calcular $p_2(1.96)$ se tiene $x_k = x_3 = 1.5$. Entonces $s = (1.96 - 1.5)/0.5 = 0.92$.

$$\begin{array}{ll} \gamma_0 = 1, & p_0(x) = f_3 = 1.2247 \\ \gamma_1 = 1(0.92) = 0.92, & p_1(x) = 1.2247 + .1895(.92) = 1.39904 \\ \gamma_2 = 0.92(0.92-1)/2 = -.0368, & p_2(x) = 1.39904 - .0226(-0.0368) \\ & p_2(x) = 1.399872 \end{array}$$

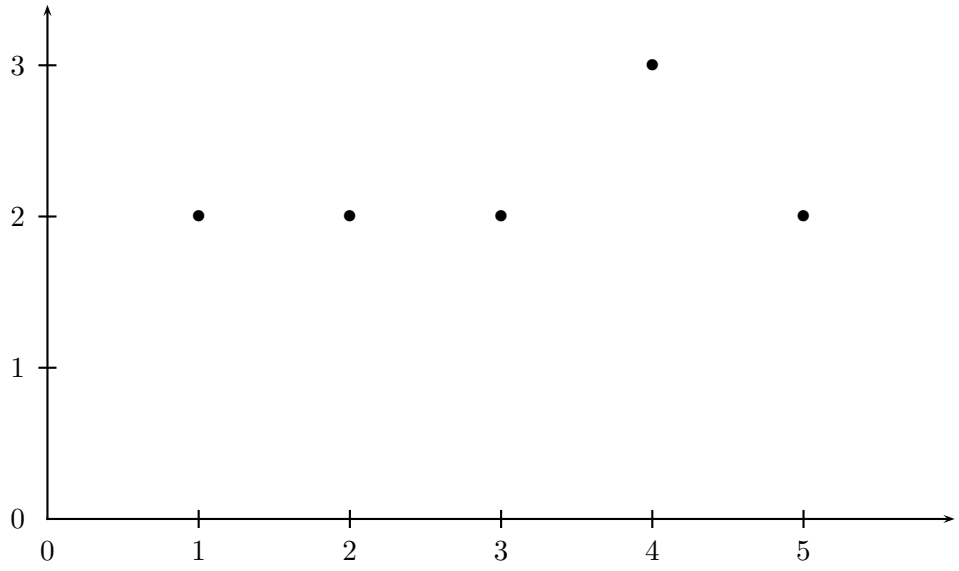


Figura 5.7: Puntos o datos iniciales

5.5 Trazadores cúbicos, interpolación polinomial por trozos, *splines*

Dados $n+1$ puntos, al utilizar diferencias divididas o diferencias finitas, cuando se desea interpolar por un polinomio de grado m en un valor t, se escoge el mejor conjunto de puntos (x_k, y_k), (x_{k+1}, y_{k+1}), ..., (x_{k+m}, y_{k+m}), para obtener el valor $p_m(t)$. Sin embargo este método presenta un gran inconveniente cuando hay que interpolar en muchos valores t. Consideremos los siguientes puntos:

$$(1,2),\ \ (2,2),\ \ (3,2),\ \ (4,3),\ \ (5,2).$$

Para interpolar por polinomios de orden 2, si $t < 2.5$ se utilizan los puntos $(1,2)$, $(2,2)$ y $(3,2)$. Entonces, por ejemplo, $p_2(2.49) = 2$. Si $2.5 < t < 3.5$, se utilizan los puntos $(2,2)$, $(3,2)$ y $(4,3)$. Después de algunos cálculos se obtiene $p_2(2.51) = 1.87505$. Para $t = 2.501$ se obtiene $p_2(2.501) = 1.8750005$. El límite de $p_2(t)$, cuando $t \to 2.5^+$, es 1.875. Esto nos muestra una discontinuidad. En $t = 3.5$ también se presenta una discontinuidad.

Estas discontinuidades se pueden evitar utilizando en el intervalo $[1,3]$ un polinomio $p_2(t)$ y en el intervalo $[3,5]$ otro polinomio $p_2(t)$.

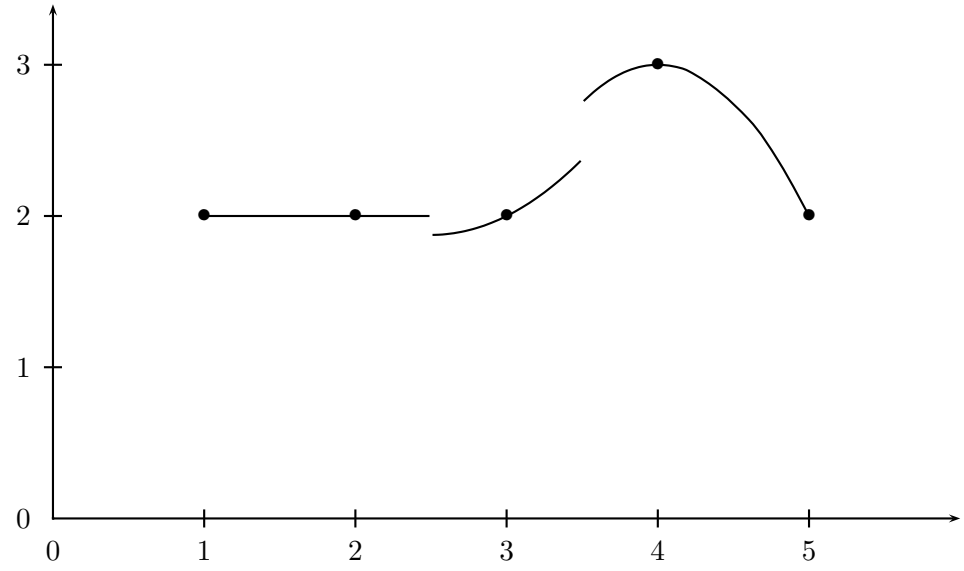


Figura 5.8: Interpolación cuadrática por trozos no continua

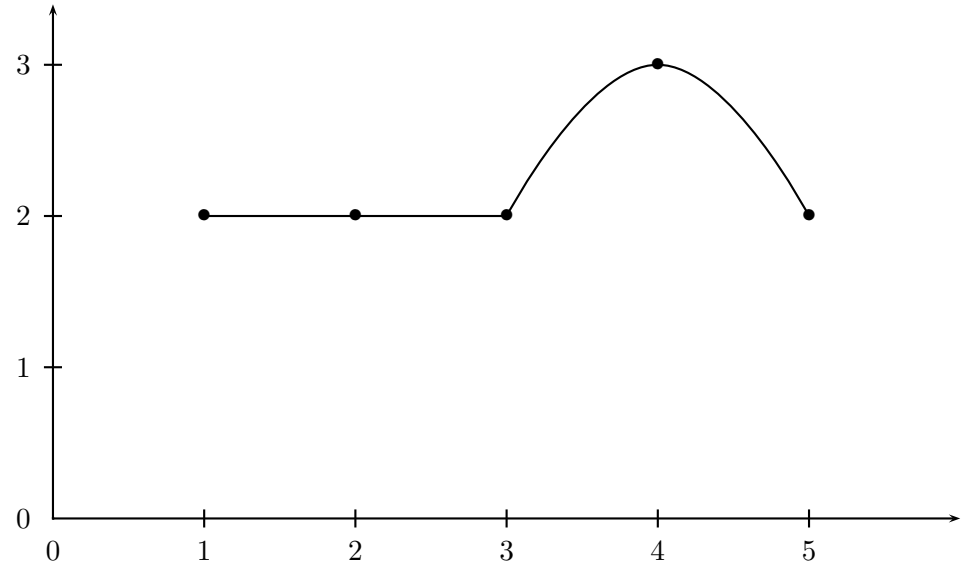


Figura 5.9: Interpolación cuadrática por trozos continua

Obviamente ya no hay discontinuidades pero la gráfica no es suave, es decir, la función interpolante no es diferenciable.

Los trazadores cúbicos ("splines" cúbicos)

remedian este inconveniente. En cada intervalo $[x_i, x_{i+1}]$ se utiliza un polinomio cúbico y los coeficientes de cada polinomio se escogen para que en los puntos x_i haya continuidad, diferenciabilidad y doble diferenciabilidad.

Dados $n+1$ puntos (x_0, y_0), (x_1, y_1), (x_2, y_2), ... (x_n, y_n), con

$$x_o < x_1 < x_2 < \cdots < x_n,$$

el trazador cúbico se define así:

$$S(x) = \begin{cases} S_0(x) & \text{si} \quad x \in [x_0, x_1] \\ S_1(x) & \text{si} \quad x \in [x_1, x_2] \\ \vdots & \\ S_{n-1}(x) & \text{si} \quad x \in [x_{n-1}, x_n] \end{cases} \tag{5.32}$$

En cada uno de los n intervalos, $S_i(x)$ es un polinomio cúbico.

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \quad i = 0, 1, ..., n-1. \tag{5.33}$$

Conocer $S(x)$ quiere decir conocer $4n$ coeficientes: a_i, b_i, c_i, d_i, para $i = 0, 1, ..., n-1$.

Se requiere que $S(x)$ pase por los puntos, y que sea doblemente diferenciable. Los problemas se pueden presentar en los extremos de los intervalos. Entonces,

$$\begin{aligned} S(x_i) &= y_i, \quad i = 0, ..., n \\ S_i(x_{i+1}) &= S_{i+1}(x_{i+1}), \quad i = 0, ..., n-2 \\ S_i'(x_{i+1}) &= S_{i+1}'(x_{i+1}), \quad i = 0, ..., n-2 \\ S_i''(x_{i+1}) &= S_{i+1}''(x_{i+1}), \quad i = 0, ..., n-2 \end{aligned}$$

Sea $h_j = x_{j+1} - x_j$, el tamaño del intervalo $[x_j, x_{j+1}]$. Las condiciones

anteriores se convierten en:

$$\begin{aligned} S_i(x_i) = d_i = y_i \qquad & i = 0, ..., n-1, \\ S_{n-1}(x_n) = a_{n-1}h_{n-1}^3 + b_{n-1}h_{n-1}^2 + c_{n-1}h_{n-1} + d_{n-1} = y_n & \\ a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i = d_{i+1} \qquad & i = 0, ..., n-2, \\ 3a_i h_i^2 + 2b_i h_i + c_i = c_{i+1} \qquad & i = 0, ..., n-2, \\ 6a_i h_i + 2b_i = 2b_{i+1} \qquad & i = 0, ..., n-2. \end{aligned}$$

Sea $d_n := y_n$ una variable adicional. Esta variable se utilizará únicamente en las fórmulas intermedias, pero no aparece en las fórmulas finales.

$$d_i = y_i \qquad i = 0, ..., n, \tag{5.34}$$

$$a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i = d_{i+1} \qquad i = 0, ..., n-1, \tag{5.35}$$

$$3a_i h_i^2 + 2b_i h_i + c_i = c_{i+1} \qquad i = 0, ..., n-2, \tag{5.36}$$

$$3a_i h_i + b_i = b_{i+1} \qquad i = 0, ..., n-2. \tag{5.37}$$

De (5.37):

$$a_i = \frac{b_{i+1} - b_i}{3h_i} \tag{5.38}$$

Reemplazando (5.38) en (5.35):

$$\begin{aligned} \frac{h_i^2}{3}(b_{i+1} - b_i) + b_i h_i^2 + c_i h_i + d_i &= d_{i+1} \\ \frac{h_i^2}{3}(b_{i+1} + 2b_i) + c_i h_i + d_i &= d_{i+1} \end{aligned} \tag{5.39}$$

Reemplazando (5.38) en (5.36):

$$\begin{aligned} (b_{i+1} - b_i)h_i + 2b_i h_i + c_i &= c_{i+1} \\ (b_{i+1} + b_i)h_i + c_i &= c_{i+1} \end{aligned} \tag{5.40}$$

Despejando c_i de (5.39):

$$c_i = \frac{1}{h_i}(d_{i+1} - d_i) - \frac{h_i}{3}(2b_i + b_{i+1}) \tag{5.41}$$

Cambiando i por $i - 1$:

$$c_{i-1} = \frac{1}{h_{i-1}}(d_i - d_{i-1}) - \frac{h_{i-1}}{3}(2b_{i-1} + b_i) \tag{5.42}$$

Cambiando i por $i-1$ en (5.40):

$$(b_i + b_{i-1})h_{i-1} + c_{i-1} = c_i \tag{5.43}$$

Reemplazando (5.41) y (5.42) en (5.43):

$$(b_i + b_{i-1})h_{i-1} + \frac{1}{h_{i-1}}(d_i - d_{i-1}) - \frac{h_{i-1}}{3}(2b_{i-1} + b_i) = \frac{1}{h_i}(d_{i+1} - d_i) - \frac{h_i}{3}(2b_i + b_{i+1})$$

Las variables d_i son en realidad constantes ($d_i = y_i$). Dejando al lado izquierdo las variables b_j y al lado derecho los términos independientes, se tiene:

$$\frac{h_{i-1}}{3}\, b_{i-1} + \left(\frac{2h_{i-1}}{3} + \frac{2h_i}{3}\right) b_i + \frac{h_i}{3}\, b_{i+1} = \frac{1}{h_{i-1}}(d_{i-1} - d_i) + \frac{1}{h_i}(d_{i+1} - d_i).$$

Multiplicando por 3:

$$h_{i-1}b_{i-1} + 2(h_{i-1} + h_i)b_i + h_i b_{i+1} = \frac{3}{h_{i-1}}(d_{i-1} - d_i) + \frac{3}{h_i}(-d_i + d_{i+1}). \tag{5.44}$$

La igualdad anterior es válida para $i = 1, ..., n-2$. Es decir, hay $n-2$ ecuaciones con n incógnitas. El sistema se completa según las condiciones de frontera. Hay dos clases de condiciones sobre $S(x)$. La primera clase se conoce con el nombre de condiciones de **frontera libre o natural**: en los extremos la curvatura es nula, o sea, $S''(x_0) = 0$ y $S''(x_n) = 0$,

$$\begin{aligned} S_0''(x_0) &= 0, \\ S_{n-1}''(x_n) &= 0. \end{aligned} \tag{5.45}$$

En la segunda clase de condiciones de frontera, **frontera sujeta**, se supone conocida la pendiente de $S(x)$ en los extremos:

$$\begin{aligned} S_0'(x_0) &= f'(x_0), \\ S_{n-1}'(x_n) &= f'(x_n). \end{aligned} \tag{5.46}$$

Al explicitar las condiciones de frontera libre se tiene:

$$\begin{aligned} S_0''(x) &= 6a_0(x - x_0) + 2b_0 \\ S_{n-1}''(x) &= 6a_{n-1}(x - x_{n-1}) + 2b_{n-1} \\ S_0''(x_0) &= 2b_0 = 0 && (5.47) \\ S_{n-1}''(x_n) &= 3a_{n-1}h_{n-1} + b_{n-1} = 0. && (5.48) \end{aligned}$$

Además del resultado anterior, $b_0 = 0$, se puede introducir una variable adicional $b_n = 0$. Esto permite que la ecuación (5.44) se pueda aplicar para $i = n - 1$. Recuérdese que ya se introdujo $d_n = y_n$ y que para todo i se tiene $d_i = y_i$. Entonces se tiene un sistema de $n + 1$ ecuaciones con $n + 1$ incógnitas, escrito de la forma

$$Ab = \zeta, \tag{5.49}$$

donde

$$A = \begin{bmatrix} 1 & 0 & 0 & & & & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & & & \\ 0 & h_1 & 2(h_1+h_2) & h_2 & & & \\ 0 & 0 & h_2 & 2(h_2+h_3) & h_3 & & \\ & & & & & & \\ & & & & & & \\ 0 & 0 & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & 0 & & & & 0 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}, \qquad \zeta = \begin{bmatrix} 0 \\ \frac{3}{h_0}(y_0 - y_1) + \frac{3}{h_1}(-y_1 + y_2) \\ \frac{3}{h_1}(y_1 - y_2) + \frac{3}{h_2}(-y_2 + y_3) \\ \vdots \\ \frac{3}{h_{n-2}}(y_{n-2} - y_{n-1}) + \frac{3}{h_{n-1}}(-y_{n-1} + y_n) \\ 0 \end{bmatrix}.$$

El sistema (5.49) tiene dos características importantes: es tridiagonal, lo cual facilita su solución; la matriz A es de diagonal estrictamente dominante, lo cual garantiza que A es invertible y que la solución existe y es única.

Una vez conocidos los valores b_0, b_1, ..., b_{n-1}, b_n, se puede aplicar (5.41) para calcular los c_i:

$$c_i = \frac{1}{h_i}(y_{i+1} - y_i) - \frac{h_i}{3}(2b_i + b_{i+1}), \quad i = 0, ..., n - 1. \tag{5.50}$$

Como b_n existe y vale 0, la ecuación (5.38) se puede aplicar aún para $i = n-1$.

$$a_i = \frac{b_{i+1} - b_i}{3h_i}, \quad i = 0, ..., n-1. \tag{5.51}$$

Obsérvese que para $i = n-1$, la igualdad $a_{n-1} = (0 - b_{n-1})/(3h_{n-1})$ coincide con la segunda condición de frontera (5.48). El orden de aplicación de las fórmulas es el siguiente:

- $d_i = y_i, \quad i = 0, ..., n-1.$
- Obtener b_0, b_1, ..., b_n resolviendo (5.49).
  En particular $b_0 = 0$ y $b_n = 0$.
- Para $i = 0, ..., n-1$ calcular c_i según (5.50).
- Para $i = 0, ..., n-1$ calcular a_i según (5.51).

Ejemplo 5.10. Construir el trazador cúbico para los puntos $(1, 2)$, $(2, 2)$, $(3, 2)$, $(4, 3)$ y $(5, 2)$.

De manera inmediata $d_0 = 2$, $d_1 = 2$, $d_2 = 2$ y $d_3 = 3$. Adicionalmente $d_4 = 2$. En este ejemplo $h_0 = h_1 = h_2 = h_3 = 1$. El sistema que permite obtener los b_i es:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 3 \\ -6 \\ 0 \end{bmatrix}.$$

Al resolver el sistema se obtiene $b_0 = 0$ (obvio), $b_1 = -0.321429$, $b_2 = 1.285714$, $b_3 = -1.821429$ y $b_4 = 0$ (también obvio). El cálculo de los otros coeficientes da:

$$\begin{aligned} c_0 &= \ \ 0.107143 \\ c_1 &= -0.214286 \\ c_2 &= \ \ 0.75 \\ c_3 &= \ \ 0.214286 \\ a_0 &= -0.107143 \\ a_1 &= \ \ 0.535714 \\ a_2 &= -1.035714 \\ a_3 &= \ \ 0.607143. \end{aligned}$$

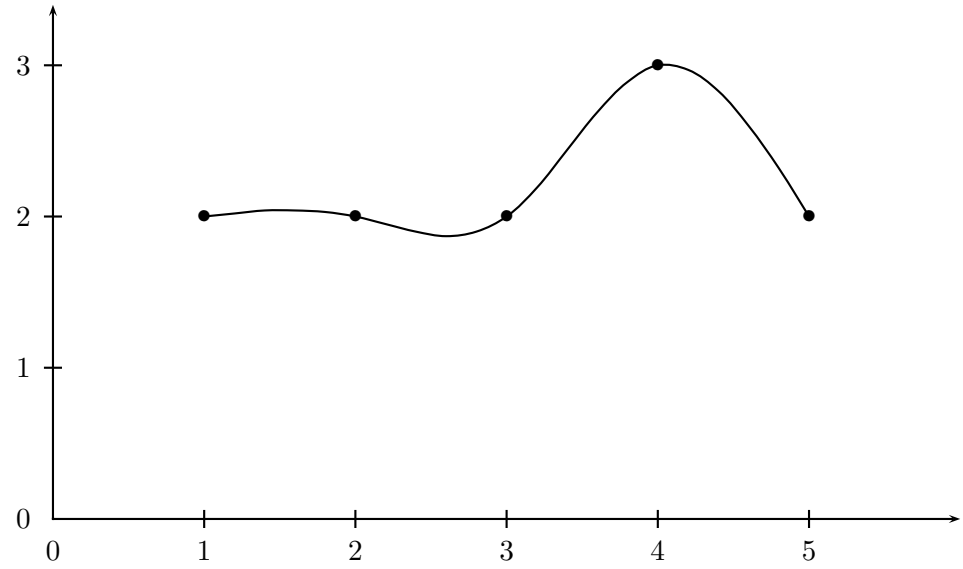


Figura 5.10: Interpolación con trazadores cúbicos o splines

Entonces

$$S_0(x) = -0.107143(x-1)^3 + 0(x-1)^2 + 0.107143(x-1) + 2$$
$$S_1(x) = 0.535714(x-2)^3 - 0.321429(x-2)^2 - 0.214286(x-2) + 2$$
$$S_2(x) = -1.035714(x-3)^3 + 1.285714(x-3)^2 + 0.75(x-3) + 2$$
$$S_3(x) = 0.607143(x-4)^3 - 1.821429(x-4)^2 + 0.214286(x-4) + 3\,.$$

5.6 Aproximación por mínimos cuadrados

Cuando hay muchos puntos no es conveniente buscar un único polinomio o una función que pase exactamente por todos los puntos. Entonces hay dos soluciones: la primera, vista anteriormente, es hacer interpolación por grupos pequeños de puntos. Para muchos casos es una solución muy buena. Sin embargo, en algunas ocasiones se desea una función que sirva para todos los puntos. La segunda solución consiste en obtener una sola función $\tilde{f}$ que, aunque no pase por todos los puntos, pase relativamente cerca de todos. Este es el enfoque de la aproximación por mínimos cuadrados.

Se supone que hay m puntos (x_1, y_1), (x_2, y_2), ..., (x_m, y_m) y que los x_i son todos diferentes. La función $\tilde{f}$, que se desea construir, debe ser combinación

lineal de n funciones llamadas funciones de la base. Supongamos que estas funciones son φ_1, φ_2, ..., φ_n. Entonces,

$$\tilde{f}(x) = a_1\varphi_1(x) + a_2\varphi_2(x) + \cdots + a_n\varphi_n(x).$$

Como las funciones de la base son conocidas, para conocer $\tilde{f}$ basta conocer los escalares a_1, a_2, ..., a_n.

Como se supone que hay muchos puntos (m grande) y como se desea que $\tilde{f}$ sea sencilla, es decir, n es relativamente pequeño, entonces se debe tener que $m \geq n$.

Las funciones de la base deben ser linealmente independientes. Los escalares a_1, a_2, ..., a_n se escogen de tal manera que $\tilde{f}(x_i) \approx y_i$, para $i = 1, 2, ..., m$. Entonces,

$$\begin{aligned}
a_1\varphi_1(x_1) + a_2\varphi_2(x_1) + \cdots + a_n\varphi_n(x_1) &\approx y_1 \\
a_1\varphi_1(x_2) + a_2\varphi_2(x_2) + \cdots + a_n\varphi_n(x_2) &\approx y_2 \\
a_1\varphi_1(x_3) + a_2\varphi_2(x_3) + \cdots + a_n\varphi_n(x_3) &\approx y_3 \\
&\vdots \\
a_1\varphi_1(x_m) + a_2\varphi_2(x_m) + \cdots + a_n\varphi_n(x_m) &\approx y_m.
\end{aligned}$$

Las m igualdades (aproximadas) anteriores se pueden escribir de manera matricial:

$$\begin{bmatrix}
\varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_n(x_1) \\
\varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_n(x_2) \\
\varphi_1(x_3) & \varphi_2(x_3) & \cdots & \varphi_n(x_3) \\
\vdots & & & \\
\varphi_1(x_m) & \varphi_2(x_m) & \cdots & \varphi_n(x_m)
\end{bmatrix}
\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}
\approx
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \\ \vdots \\ y_m \end{bmatrix}$$

De manera compacta se tiene

$$\Phi\, a \approx y. \tag{5.52}$$

La matriz Φ es una matriz $m \times n$ rectangular alta ($m \geq n$), a es un vector columna $n \times 1$, y es un vector columna $m \times 1$. Son conocidos la matriz Φ y el vector columna y. El vector columna a es el vector de incógnitas. Como las funciones de la base son linealmente independientes, entonces las columnas de Φ son linealmente independientes. En consecuencia, (5.52) se puede resolver por mínimos cuadrados:

$$(\Phi^{\mathrm{T}}\Phi)\, a = \Phi^{\mathrm{T}} y\,. \tag{5.53}$$

Recordemos del capítulo 11 que para resolver por mínimos cuadrados el sistema $Ax = b$, se minimiza $||Ax - b||_2^2$. Traduciendo esto al problema de aproximación por mínimos cuadrados, se tiene

$$\min \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_j \varphi_j(x_i) - y_i \right)^2 .$$

es decir,

$$\min \sum_{i=1}^{m} \left(\tilde{f}(x_i) - y_i \right)^2 .$$

Esto significa que se está buscando una función $\tilde{f}$, combinación lineal de las funciones de la base, tal que minimiza la suma de los cuadrados de las distancias entre los puntos $(x_i, \tilde{f}(x_i))$ y (x_i, y_i).

Ejemplo 5.11. Dadas las funciones $\varphi_1(x) = 1$, $\varphi_2(x) = x$, $\varphi_3(x) = x^2$, encontrar la función $\tilde{f}$ que aproxima por mínimos cuadrados la función dada por los puntos $(0, 0.55)$, $(1, 0.65)$, $(1.5, 0.725)$, $(2, 0.85)$, $(3, 1.35)$.

Como las funciones de la base son 1, x, x^2, en realidad se está buscando aproximar por mínimos cuadrados por medio de un parábola. El sistema inicial es

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1.5 & 2.25 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \approx \begin{bmatrix} 0.55 \\ 0.65 \\ 0.725 \\ 0.85 \\ 1.35 \end{bmatrix}$$

Las ecuaciones normales dan:

$$\begin{bmatrix} 5 & 7.5 & 16.25 \\ 7.5 & 16.25 & 39.375 \\ 16.25 & 39.375 & 103.0625 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 4.1250 \\ 7.4875 \\ 17.8313 \end{bmatrix}$$

La solución es:

$$a = \begin{bmatrix} 0.56 \\ -0.04 \\ 0.10 \end{bmatrix} , \qquad \tilde{f}(x) = 0.56 - 0.04x + 0.1x^2.$$

$$\begin{bmatrix} \tilde{f}(x_1) \\ \tilde{f}(x_2) \\ \tilde{f}(x_3) \\ \tilde{f}(x_4) \\ \tilde{f}(x_5) \end{bmatrix} = \Phi\, a = \begin{bmatrix} 0.56 \\ 0.62 \\ 0.725 \\ 0.88 \\ 1.34 \end{bmatrix} , \qquad y = \begin{bmatrix} 0.55 \\ 0.65 \\ 0.725 \\ 0.85 \\ 1.35 \end{bmatrix} \quad \diamond$$

Ejercicios

5.1 Halle, resolviendo el sistema de ecuaciones, el polinomio de interpolación que pasa por los puntos

$$(1, -5),$$
$$(2, -4),$$
$$(4, \ \ 4).$$

5.2 Halle, por medio de los polinomios de Lagrange, el polinomio de interpolación que pasa por los puntos del ejercicio anterior.

5.3 Halle el polinomio de interpolación que pasa por los puntos

$$(-1, -5),$$
$$(1, -5),$$
$$(2, -2),$$
$$(4, \ 40).$$

5.4 Halle el polinomio de interpolación que pasa por los puntos

$$(-1, \ \ 10),$$
$$(1, \ \ \ 8),$$
$$(2, \ \ \ 4),$$
$$(4, -10).$$

5.5 Considere los puntos

$$(0.10, 11.0000),$$
$$(0.13, \ 8.6923),$$
$$(0.16, \ 7.2500),$$
$$(0.20, \ 6.0000),$$
$$(0.26, \ 4.8462),$$
$$(0.40, \ 3.5000),$$
$$(0.32, \ 4.1250),$$
$$(0.50, \ 3.0000).$$

Construya la tabla de diferencias dividas hasta el orden 3. Obtenga $p_2(0.11)$, $p_2(0.08)$, $p_2(0.25)$, $p_2(0.12)$, $p_2(0.33)$, $p_2(0.6)$, $p_3(0.25)$, $p_3(0.33)$, $p_3(0.6)$.

5.6 Considere los puntos

$$(0.05, 21.0000),$$
$$(0.10, 11.0000),$$
$$(0.15, \ 7.6667),$$
$$(0.20, \ 6.0000),$$
$$(0.25, \ 5.0000),$$
$$(0.30, \ 4.3333),$$
$$(0.35, \ 3.8571),$$
$$(0.40, \ 3.5000).$$

Construya la tabla de diferencias divididas hasta el orden 3. Calcule $p_2(0.11)$, $p_2(0.08)$, $p_2(0.25)$, $p_2(0.12)$, $p_2(0.33)$, $p_2(0.6)$, $p_3(0.25)$, $p_3(0.33)$, $p_3(0.6)$.

5.7 Considere los mismos puntos del ejercicio anterior. Construya la tabla de diferencias finitas hasta el orden 3. Halle $p_2(0.11)$, $p_2(0.08)$, $p_2(0.25)$, $p_2(0.12)$, $p_2(0.33)$, $p_2(0.6)$, $p_3(0.25)$, $p_3(0.33)$, $p_3(0.6)$.

5.8 Considere los puntos

$$(0.05, 2.0513),$$
$$(0.10, 2.1052),$$
$$(0.15, 2.1618),$$
$$(0.20, 2.2214),$$
$$(0.25, 2.2840),$$
$$(0.30, 2.3499),$$
$$(0.35, 2.4191),$$
$$(0.40, 2.4918).$$

Obtenga la recta de aproximación por mínimos cuadrados.

5.9 Considere los mismos puntos del ejercicio anterior. Obtenga la parábola de aproximación por mínimos cuadrados.

5.10 Considere los mismos puntos de los dos ejercicios anteriores. Use otra base y obtenga la correspondiente función de aproximación por mínimos cuadrados.

6

Integración y diferenciación

6.1 Integración numérica

Esta técnica sirve para calcular el valor numérico de una integral definida, es decir, parar obtener el valor

$$I = \int_a^b f(x)dx.$$

En la mayoría de los casos no se puede calcular el valor exacto I; simplemente se calcula $\tilde{I}$ aproximación de I.

De todas maneras primero se debe tratar de hallar la antiderivada. Cuando esto sea imposible o muy difícil, entonces se recurre a la integración numérica. Por ejemplo, calcular una aproximación de

$$\int_{0.1}^{0.5} e^{x^2} dx.$$

En este capítulo hay ejemplos de integración numérica con funciones cuya antiderivada es muy fácil de obtener y para los que no se debe utilizar la integración numérica; se usan solamente para comparar el resultado aproximado con el valor exacto.

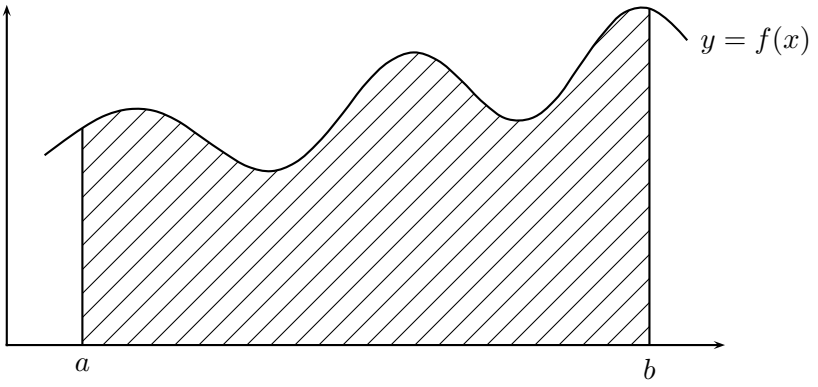


Figura 6.1: Integral definida

6.2 En Scilab

Para obtener una aproximación del valor de una integral definida, por ejemplo,

$$\int_{0.1}^{0.5} e^{-x^2} dx$$

se utiliza `intg`. Para eso es necesario definir en Scilab la función que se va a integrar. Puede ser, directamente en el ambiente Scilab:

```
deff('[y] = f53(x)', 'y = exp(-x*x)')
I = intg(0.1, 0.5, f53)
```

También se puede definir una función en un archivo `.sci`

```
function fx = f57(x)
  fx = exp(-x*x)
endfunction
```

y después de cargarla, dar la orden

```
I = intg(0.1, 0.5, f57)
```

También se puede utilizar la función `integrate`:

```
I = integrate('exp(-x*x)', 'x', 0.1, 0.5)
```

Aunque Scilab es muy bueno, no es perfecto. La utilización de `intg` o `integrate` no funciona bien (versión 5.1) para

$$\int_0^{2\pi} \text{sen}(x)\, dx\,.$$

Algunas veces no se conoce una expresión de la función f, pero se conoce una tabla de valores $(x_i, f(x_i))$, o simplemente una tabla de valores (x_i, y_i). Supongamos, así lo requiere Scilab, que la lista de valores (x_1, y_1), ..., (x_n, y_n) está ordenada de manera creciente de acuerdo a los x_i, o sea, $x_1 < x_2 < \cdots < x_n$.

Para obtener el valor aproximado de la integral, entre x_1 y x_n, de la función f (representada por los valores (x_i, y_i)), es necesario tener dos vectores con los valor x_i y y_i, y utilizar la función `inttrap`, que utiliza la fórmula del trapecio en cada subintervalo.

```
x = [0.1 0.15 0.2 0.25 0.3 0.4 0.5]'
y = [ 0.9900 0.9778 0.9608 0.9394 0.9139 0.8521 0.7788]'
I = inttrap(x, y)
```

Para los mismos parámetros `x , y`, se puede utilizar la función `intsplin` que utiliza trazadores cúbicos (*splines*).

```
x = [0.1 0.15 0.2 0.25 0.3 0.4 0.5]'
y = [ 0.9900 0.9778 0.9608 0.9394 0.9139 0.8521 0.7788]'
I = intsplin(x, y)
```

6.3 Fórmula del trapecio

La fórmula del trapecio, como también la fórmula de Simpson, hace parte de las fórmulas de Newton-Cotes. Sean $n+1$ valores igualmente espaciados $a = x_0$, x_1, x_2, ..., $x_n = b$, donde

$$x_i = a + ih\,,\ \ i = 0, 1, 2, ..., n\,,\ \ h = \frac{b-a}{n}\,,$$

y supongamos conocidos $y_i = f(x_i)$. Supongamos además que n es un múltiplo de m, $n = km$. La integral $\int_{x_0}^{x_n} f(x)dx$ se puede separar en inter-

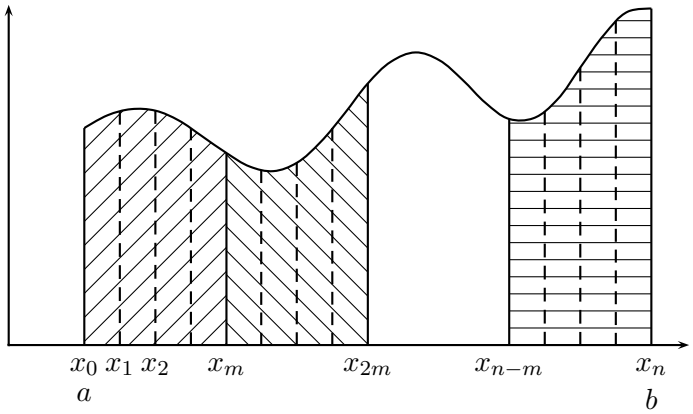


Figura 6.2: División en subintervalos

valos más pequeños:

$$\int_{x_0}^{x_n} f(x)dx = \int_{x_0}^{x_m} f(x)dx + \int_{x_m}^{x_{2m}} f(x)dx + \cdots + \int_{x_{n-m}}^{x_n} f(x)dx.$$

En el intervalo $[x_0, x_m]$ se conocen los puntos (x_0, y_0), (x_1, y_1), ..., (x_m, y_m) y se puede construir el polinomio de interpolación de Lagrange $p_m(x)$. Entonces la integral $\int_{x_0}^{x_m} f(x)dx$ se aproxima por la integral de p_m,

$$\int_{x_0}^{x_m} f(x)dx \approx \int_{x_0}^{x_m} p_m(x)dx.$$

Para $m = 1$ se tiene la fórmula del trapecio. Su deducción es mucho más sencilla si se supone que $x_0 = 0$. Esto equivale a hacer el cambio de variable $x' = x - x_0$.

$$\begin{aligned} p_1(x) &= y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}, \\ p_1(x) &= y_0 \frac{x - h}{-h} + y_1 \frac{x}{h}, \\ p_1(x) &= y_0 + x\,\frac{y_1 - y_0}{h}. \end{aligned}$$

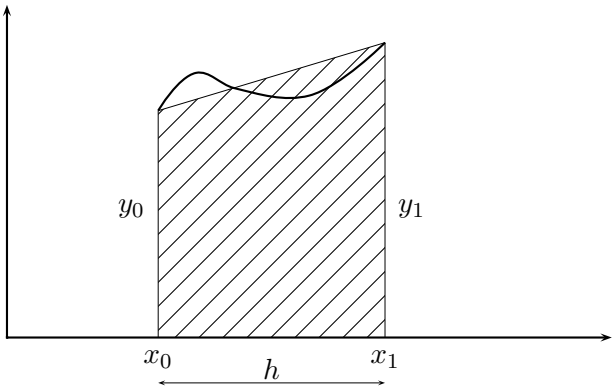


Figura 6.3: Fórmula del trapecio

Entonces

$$\begin{aligned}\int_{x_0}^{x_1} p_1(x)dx &= \int_0^h (y_0 + x\,\frac{y_1 - y_0}{h})dx \\ &= y_0 h + \frac{h^2}{2}\frac{y_1 - y_0}{h}, \\ &= h(\frac{y_0}{2} + \frac{y_1}{2}), \\ \int_{x_0}^{x_1} f(x)dx &\approx h\,\frac{y_0 + y_1}{2}. \end{aligned} \tag{6.1}$$

De la fórmula (6.1) o de la gráfica se deduce naturalmente el nombre de fórmula del trapecio.

Ejemplo 6.1.

$$\int_0^{0.2} e^x dx \approx 0.2(\frac{1}{2}e^0 + \frac{1}{2}e^{0.2}) = 0.22214028\,.\ \diamond$$

Aplicando la fórmula del trapecio a cada uno de los intervalos $[x_{i-1}, x_i]$ se tiene:

$$\begin{aligned}\int_{x_0}^{x_1} f(x)dx &\approx h(\frac{y_0}{2} + \frac{y_1}{2}), \\ \int_{x_1}^{x_2} f(x)dx &\approx h(\frac{y_1}{2} + \frac{y_2}{2}), \\ \vdots \quad &\approx \quad \vdots \\ \int_{x_{n-1}}^{x_n} f(x)dx &\approx h(\frac{y_{n-1}}{2} + \frac{y_n}{2}).\end{aligned}$$

$$\int_{x_0}^{x_n} f(x)dx \approx h(\frac{y_0}{2} + \frac{y_1}{2} + \frac{y_1}{2} + \frac{y_2}{2} + \cdots \frac{y_{n-1}}{2} + \frac{y_n}{2}),$$
$$\int_{x_0}^{x_n} f(x)dx \approx h(\frac{y_0}{2} + y_1 + y_2 + \cdots + y_{n-2} + y_{n-1} + \frac{y_n}{2}), \qquad (6.2)$$
$$\int_{x_0}^{x_n} f(x)dx \approx h(\frac{y_0}{2} + \sum_{i=1}^{n-1} y_i + \frac{y_n}{2}).$$

Ejemplo 6.2.

$$\int_0^{0.8} e^x dx \approx 0.2(\frac{1}{2}e^0 + e^{0.2} + e^{0.4} + e^{0.6} + \frac{1}{2}e^{0.8}) = 1.22962334\,. \ \diamond$$

6.3.1 Errores local y global

El error local de la fórmula del trapecio es el error proveniente de la fórmula (6.1).

$$\begin{aligned} e_{\text{loc}} &= I_{\text{loc}} - \tilde{I}_{\text{loc}}, \\ e_{\text{loc}} &= \int_{x_0}^{x_1} f(x)dx - h(\frac{y_0}{2} + \frac{y_1}{2}) \\ &= \int_{x_0}^{x_1} f(x)dx - \int_{x_0}^{x_1} p_1(x)dx \\ &= \int_{x_0}^{x_1} (f(x) - p_1(x))dx\,. \end{aligned}$$

Utilizando la fórmula del error para la interpolación polinómica 5.6,

$$e_{\text{loc}} = \int_{x_0}^{x_1} \frac{(x-x_0)(x-x_1)}{2} f''(\xi_x)dx\,,\ \xi_x \in [x_0, x_1].$$

El teorema del valor medio para integrales dice: *Sean f continua en $[a,b]$, g integrable en $[a,b]$, g no cambia de signo en $[a,b]$, entonces*

$$\int_a^b f(x)g(x)dx = f(c)\int_a^b g(x)dx$$

para algún c en $[a,b]$.

Teniendo en cuenta que $(x-x_0)(x-x_1) \leq 0$ en el intervalo $[x_0,x_1]$ y aplicando el teorema del valor medio para integrales, existe $z \in [x_0,x_1]$ tal que

$$e_{\text{loc}} = \frac{f''(z)}{2}\int_{x_0}^{x_1} (x-x_0)(x-x_1)dx\,,\ z \in [x_0,x_1].$$

Mediante el cambio de variable $t = x - x_0$, $dt = dx$,

$$\begin{aligned} e_{\text{loc}} &= \frac{f''(z)}{2} \int_0^h t(t-h)dt\,,\ z \in [x_0, x_1], \\ &= \frac{f''(z)}{2}(-\frac{h^3}{6})\,,\ z \in [x_0, x_1], \\ e_{\text{loc}} &= -h^3\,\frac{f''(z)}{12}\,,\ z \in [x_0, x_1]. \end{aligned} \tag{6.3}$$

La fórmula anterior, como muchas de las fórmulas de error, sirve principalmente para obtener cotas del error cometido.

$$|e_{\text{loc}}| \leq \frac{h^3}{12} M\,, \quad M = \max\{|f''(z)| : z \in [x_0, x_1]\}. \tag{6.4}$$

En el ejemplo 6.1, $f''(x) = e^x$, $\max\{|f''(z)| : z \in [0,\, 0.2]\} = 1.22140276$, luego el máximo error que se puede cometer, en valor absoluto, es $(0.2)^3 \times 1.22140276/12 = 8.1427 \cdot 10^{-4}$. En este ejemplo, se conoce el valor exacto $I = e^{0.2} - 1 = 0.22140276$, luego $|e| = 7.3752 \cdot 10^{-4}$.

En algunos casos, la fórmula del error permite afinar un poco más. Si $f''(x) > 0$ (f estrictamente convexa) en $[x_0, x_1]$ y como $I = \tilde{I} + e_{\text{loc}}$, entonces la fórmula del trapecio da un valor aproximado pero superior al exacto.

En el mismo ejemplo, $f''(x)$ varía en el intervalo $[1,\, 1.22140276]$ cuando $x \in [0,\, 0.2]$. Luego

$$e_{\text{loc}} \in [-0.00081427,\, -0.00066667],$$

entonces

$$I \in [0.22132601, 0.22147361].$$

El error global es el error correspondiente al hacer la aproximación de la integral sobre todo el intervalo $[x_0, x_n]$, o sea, el error en la fórmula 6.2,

$$\begin{aligned} e_{\text{glob}} &= \int_{x_0}^{x_n} f(x)dx - h(\frac{y_0}{2} + y_1 + y_2 + \cdots + y_{n-2} + y_{n-1} + \frac{y_n}{2}) \\ &= \sum_{i=1}^{n} (-\frac{f''(z_i)\,h^3}{12})\,, \quad z_i \in [x_{i-1}, x_i] \\ &= -\frac{h^3}{12} \sum_{i=1}^{n} f''(z_i)\,, \quad z_i \in [x_{i-1}, x_i] \end{aligned}$$

Sean

$$M_1 = \min\{f''(x) : x \in [a,b]\}\,, \quad M_2 = \max\{f''(x) : x \in [a,b]\}.$$

Entonces

$$\begin{aligned} M_1 \le f''(z_i) \le M_2\,, \quad \forall i \\ nM_1 \le \sum_{i=1}^{n} f''(z_i) \le nM_2\,, \\ M_1 \le \frac{1}{n}\sum_{i=1}^{n} f''(z_i) \le M_2\,. \end{aligned}$$

Si $f \in C^2_{[a,b]}$, entonces, aplicando el teorema del valor intermedio a f'', existe $\xi \in [a,b]$ tal que

$$f''(\xi) = \frac{1}{n}\sum_{i=1}^{n} f''(z_i)\,.$$

Entonces

$$e_{\text{glob}} = -\frac{h^3}{12} n f''(\xi)\,, \quad \xi \in [a,b].$$

Como $h = (b-a)/n$, entonces $n = (b-a)/h$.

$$e_{\text{glob}} = -h^2\,\frac{(b-a)f''(\xi)}{12}\,, \quad \xi \in [a,b]. \tag{6.5}$$

6.4 Fórmula de Simpson

Es la fórmula de Newton-Cotes para $m = 2$,

$$\int_{x_0}^{x_2} f(x)dx \approx \int_{x_0}^{x_2} p_2(x)dx.$$

El polinomio de interpolación $p_2(x)$ se construye a partir de los puntos (x_0, y_0), (x_1, y_1), (x_2, y_2). Para facilitar la deducción de la fórmula, supongamos que p_2 es el polinomio de interpolación que pasa por los puntos $(0, y_0)$, (h, y_1), $(2h, y_2)$. Entonces

$$\begin{aligned} p_2(x) &= y_0\,\frac{(x-h)(x-2h)}{(0-h)(0-2h)} + y_1\,\frac{(x-0)(x-2h)}{(h-0)(h-2h)} + y_2\,\frac{(x-0)(x-h)}{(2h-0)(2h-h)}, \\ &= \frac{1}{2h^2}\big(y_0(x-h)(x-2h) - 2y_1\,x(x-2h) + y_2\,x(x-h)\big), \\ &= \frac{1}{2h^2}\big(x^2(y_0 - 2y_1 + y_2) + hx(-3y_0 + 4y_1 - y_2) + 2h^2 y_0\big), \end{aligned}$$

$$\int_0^{2h} p_2(x)dx = \frac{1}{2h^2}\Big(\frac{8h^3}{3}(y_0 - 2y_1 + y_2) + h\frac{4h^2}{2}(-3y_0 + 4y_1 - y_2) + 2h^2(2h)y_0\Big),$$

$$\int_0^{2h} p_2(x)dx = h(\frac{1}{3}y_0 + \frac{4}{3}y_1 + \frac{1}{3}y_2).$$

Entonces

$$\int_{x_0}^{x_2} f(x)dx \approx \frac{h}{3}(y_0 + 4y_1 + y_2) \tag{6.6}$$

Suponiendo que n es par, al aplicar la fórmula anterior a cada uno de los intervalos $[x_0, x_2]$, $[x_2, x_4]$, $[x_4, x_6]$, ..., $[x_{n-4}, x_{n-2}]$, $[x_{n-2}, x_n]$, se tiene:

$$\int_{x_0}^{x_n} f(x)dx \approx \frac{h}{3}(y_0 + 4y_1 + 2y_2 + 4y_3 + \cdots + 4y_{n-1} + y_n) \tag{6.7}$$

$$\int_{x_0}^{x_n} f(x)dx \approx \frac{h}{3}(\ y_0 + 4\sum_{j=1}^{k} y_{2j-1} + 2\sum_{j=1}^{k-1} y_{2j}\ +\ y_n\ )$$

Ejemplo 6.3.

$$\int_0^{0.8} e^x\,dx \approx \frac{0.2}{3}(e^0 + 4(e^{0.2} + e^{0.6}) + 2\,e^{0.4} + e^{0.8}) = 1.22555177\,.$$

El valor exacto, con 8 cifras decimales, es 1.22554093, entonces el error es $-0.00001084\,.$ $\diamond$

6.4.1 Errores local y global

Para facilitar la deducción del error local, consideremos la integral entre $-h$ y h. Sea $f \in C^4_{[-h,h]}$.

$$e(h) = e_{\text{loc}}(h) = \int_{-h}^{h} f(x)\,dx - \int_{-h}^{h} p_2(x)\,dx,$$

$$= \int_{-h}^{h} f(x)\,dx - \frac{h}{3}\big(f(-h) + 4f(0) + f(h)\big).$$

Sea F tal que $F'(x) = f(x)$, entonces $\int_{-h}^{h} f(x)\,dx = F(h) - F(-h)$. Al derivar con respecto a h se tiene $f(h) + f(-h)$.

$$\begin{aligned} e'(h) &= f(h) + f(-h) - \frac{1}{3}\big(f(-h) + 4f(0) + f(h)\big) \\ &\quad - \frac{h}{3}\big(-f'(-h) + f'(h)\big), \\ 3e'(h) &= 2f(h) + 2f(-h) - 4f(0) - h(f'(h) - f'(-h)). \end{aligned}$$

$$\begin{aligned} 3e''(h) &= 2f'(h) - 2f'(-h) - f'(h) + f'(-h) - h(f''(h) + f''(-h)), \\ &= f'(h) - f'(-h) - h(f''(h) + f''(-h)). \end{aligned}$$

$$\begin{aligned} 3e'''(h) &= f''(h) + f''(-h) - (f''(h) + f''(-h)) - h(f'''(h) - f'''(-h)), \\ &= -h(f'''(h) - f'''(-h)), \\ e'''(h) &= -\frac{h}{3}\ (\ f'''(h) - f'''(-h)\ ), \\ e'''(h) &= -\frac{2h^2}{3}\ \frac{f'''(h) - f'''(-h)}{2h}. \end{aligned}$$

De los resultados anteriores se ve claramente que $e(0) = e'(0) = e''(0) = e'''(0) = 0$. Además, como $f \in C^4$, entonces $f''' \in C^1$. Por el teorema del valor medio, existe $\beta \in [-h, h]$, $\beta = \alpha h$, $\alpha \in [-1, 1]$, tal que

$$\frac{f'''(h) - f'''(-h)}{2h} = f^{(4)}(\alpha h)\,,\ \ \alpha \in [-1, 1].$$

Entonces

$$e'''(h) = -\frac{2h^2}{3} f^{(4)}(\alpha h)\,,\ \ \alpha \in [-1, 1].$$

Sea

$$\begin{aligned} g_4(h) &= f^{(4)}(\alpha h). \\ e'''(h) &= -\frac{2h^2}{3}\, g_4(h). \end{aligned}$$

$$\begin{aligned} e''(h) &= \int_0^h e'''(t)\,dt + e''(0), \\ e''(h) &= -\frac{2}{3}\int_0^h t^2 g_4(t)\,dt. \end{aligned}$$

Como g_4 es continua, t^2 es integrable y no cambia de signo en $[0,h]$, se puede aplicar el teorema del valor medio para integrales,

$$e''(h) = -\frac{2}{3}\, g_4(\xi_4) \int_0^h t^2\, dt\,, \quad \xi_4 \in [0,h],$$
$$e''(h) = -\frac{2}{9}\, h^3\, g_4(\xi_4).$$

Sea

$$g_3(h) = g_4(\xi_4) = f^{(4)}(\theta_3 h)\,, \quad -1 \le \theta_3 \le 1,$$

entonces

$$e''(h) = -\frac{2}{9}\, h^3\, g_3(h).$$

De manera semejante,

$$e'(h) = \int_0^h e''(t)\, dt + e'(0),$$
$$e'(h) = -\frac{2}{9} \int_0^h t^3 g_3(t)\, dt,$$
$$e'(h) = -\frac{2}{9}\, g_3(\xi_3) \int_0^h t^3\, dt\,, \quad \xi_3 \in [0,h],$$
$$e'(h) = -\frac{1}{18}\, h^4\, g_3(\xi_3).$$

Sea

$$g_2(h) = g_3(\xi_3) = f^{(4)}(\theta_2 h)\,, \quad -1 \le \theta_2 \le 1,$$
$$e'(h) = -\frac{1}{18}\, h^4\, g_2(h).$$

$$e(h) = \int_0^h e'(t)\, dt + e(0),$$
$$e(h) = -\frac{1}{18} \int_0^h t^4 g_2(t)\, dt,$$
$$e(h) = -\frac{1}{18}\, g_2(\xi_2) \int_0^h t^4\, dt\,, \quad \xi_2 \in [0,h],$$
$$e(h) = -\frac{1}{90}\, h^5\, g_2(\xi_2),$$
$$e(h) = -\frac{h^5}{90}\, f^{(4)}(\theta_1 h)\,, \quad -1 \le \theta_1 \le 1,$$
$$e(h) = -\frac{h^5}{90}\, f^{(4)}(z)\,, \quad -h \le z \le h.$$

Volviendo al intervalo $[x_0, x_2]$,

$$e_{\text{loc}} = -h^5 \, \frac{f^{(4)}(z)}{90} \, , \; z \in [x_0, x_2]. \tag{6.8}$$

La deducción del error global se hace de manera semejante al error global en la fórmula del trapecio. Sean $n = 2k$, $M_1 = \min\{f^{(4)}(x) : x \in [a,b]\}$, $M_2 = \max\{f^{(4)}(x) : x \in [a,b]\}$.

$$\begin{aligned} e_{\text{glob}} &= \int_a^b f(x)\,dx - \Big(\frac{h}{3} (\; y_0 + 4 \sum_{j=1}^{k} y_{2j-1} + 2 \sum_{j=1}^{k-1} y_{2j} \; + \; y_n \;) \Big), \\ &= \sum_{j=1}^{k} \Big(- h^5 \, \frac{f^{(4)}(z_j)}{90} \Big) \, , \quad z_j \in [x_{2j-2}, x_{2j}], \\ &= -\frac{h^5}{90} \sum_{j=1}^{k} f^{(4)}(z_j) \end{aligned}$$

$$\begin{aligned} M_1 \le f^{(4)}(z_j) \le M_2 \, , \quad \forall j \\ kM_1 \le \sum_{j=1}^{k} f^{(4)}(z_j) \le kM_2 \, , \\ M_1 \le \frac{1}{k} \sum_{j=1}^{k} f^{(4)}(z_j) \le M_2 \, , \end{aligned}$$

Entonces, existe $\xi \in [a,b]$, tal que

$$\begin{aligned} \frac{1}{k} \sum_{j=1}^{k} f^{(4)}(z_j) &= f^{(4)}(\xi), \\ \sum_{j=1}^{k} f^{(4)}(z_j) &= k \, f^{(4)}(\xi), \\ \sum_{j=1}^{k} f^{(4)}(z_j) &= \frac{n}{2} \, f^{(4)}(\xi), \\ \sum_{j=1}^{k} f^{(4)}(z_j) &= \frac{b-a}{2h} \, f^{(4)}(\xi). \end{aligned}$$

Entonces

$$e_{\text{glob}} = -h^4 \, \frac{(b-a) f^{(4)}(\xi)}{180} \, , \quad \xi \in [a,b]. \tag{6.9}$$

La fórmula de Simpson es exacta para polinomios de grado inferior o igual a 3. El error global es del orden de h^4.

Pasando de una interpolación lineal (fórmula del trapecio) a una interpolación cuadrática (fórmula de Simpson), el error global pasa de $O(h^2)$ a $O(h^4)$, es decir, una mejora notable. Se puede ver que al utilizar interpolación cúbica se obtiene

$$\int_{x_0}^{x_3} f(x)dx = \frac{h}{8}(3y_0 + 9y_1 + 9y_2 + 3y_3) - \frac{3}{80}h^5 f^{(4)}(z)\,, \quad z \in [x_0, x_3],$$

llamada segunda fórmula de Simpson. Entonces el error local es $O(h^5)$ y el error global es $O(h^4)$. La fórmula anterior es exacta para polinomios de grado inferior o igual a 3. En resumen, la interpolación cúbica no mejora la calidad de la aproximación numérica, luego es preferible utilizar la fórmula (6.7), más sencilla y de calidad semejante.

Sin embargo, cuando se tiene una tabla fija con un número impar de subintervalos (n impar, número par de puntos), se puede aplicar la (primera) fórmula de Simpson sobre el intervalo $[x_0, x_{n-3}]$ y la segunda fórmula sobre el intervalo $[x_{n-3}, x_n]$.

6.5 Otras fórmulas de Newton-Cotes

Las fórmulas de Newton-Cotes se pueden clasificar en abiertas y cerradas. Las fórmulas del trapecio y de Simpson son casos particulares de las fórmulas cerradas. En ellas se aproxima la integral en el intervalo $[x_0, x_m]$ usando el polinomio de interpolación, de grado menor o igual a m, construido a partir de los puntos (x_0, y_0), (x_1, y_1), ..., (x_{m-1}, y_{m-1}), (x_m, y_m), igualmente espaciados en x.

$$\int_{x_0}^{x_m} f(x)dx \approx \int_{x_0}^{x_m} p_m(x)dx.$$

La siguiente tabla muestra las más importantes.

m		error
1	$\frac{h}{2}(y_0 + y_1)$	$-\frac{f''(z)}{12}\,h^3$
2	$\frac{h}{3}(y_0 + 4y_1 + y_2)$	$-\frac{f^{(4)}(z)}{90}\,h^5$
3	$\frac{3h}{8}(y_0 + 3y_1 + 3y_2 + y_3)$	$-\frac{3\,f^{(4)}(z)}{80}\,h^5$
4	$\frac{2h}{45}(7y_0 + 32y_1 + 12y_2 + 32y_3 + 7y_4)$	$-\frac{8\,f^{(6)}(z)}{945}\,h^7$

En todos los casos, $z \in [x_0, x_m]$.

6.5.1 Fórmulas de Newton-Cotes abiertas

En estas fórmulas el polinomio de interpolación se calcula sin utilizar los extremos del intervalo de integración,

$$\int_{x_0}^{x_{m+2}} f(x)dx \approx \int_{x_0}^{x_{m+2}} p_m(x)dx,$$

donde p_m, polinomio de grado menor o igual a m, se construye utilizando los puntos (x_1, y_1), (x_2, y_2), ..., (x_m, y_m), (x_{m+1}, y_{m+1}), igualmente espaciados en x.

m		error
0	$2h\,y_1$	$+\frac{f''(z)}{3}\,h^3$
1	$\frac{3h}{2}(y_1 + y_2)$	$+\frac{3\,f''(z)}{4}\,h^3$
2	$\frac{4h}{3}(2y_1 - y_2 + 2y_3)$	$+\frac{14\,f^{(4)}(z)}{45}\,h^5$
3	$\frac{5h}{24}(11y_1 + y_2 + y_3 + 11y_4)$	$+\frac{95\,f^{(4)}(z)}{144}\,h^5$

En todos los casos $z \in [x_0, x_{m+2}]$.

Ejemplo 6.4.

$$\int_0^{0.8} e^x\,dx \approx \frac{4 \times 0.2}{3}(2\,e^{0.2} - e^{0.4} + 2\,e^{0.6}) = 1.22539158\,.$$

El valor exacto, con 8 cifras decimales, es 1.22554093, entonces el error es 0.00014935 . $\diamond$

En general, las fórmulas cerradas son más precisas que las abiertas, entonces, siempre que se pueda, es preferible utilizar las fórmulas cerradas. Las fórmulas abiertas se usan cuando no se conoce el valor de la función f en los extremos del intervalo de integración; por ejemplo, en la solución numérica de algunas ecuaciones diferenciales ordinarias.

6.6 Cuadratura adaptativa

Sea $I = \int_a^b f(x)dx$ e I_n la aproximación de I por un método fijo de Newton-Cotes (trapecio, Simpson,...) utilizando n subintervalos. La fórmula que relaciona I, I_n y el error global se puede expresar así:

$$I = I_n + F(b-a)h^p f^{(q)}(\xi), \text{ para algún } \xi \in [a,b],$$

donde F, p y q dependen del método escogido; ξ depende del método, de la función f, de n y del intervalo. Entonces

$$\begin{aligned} I &= I_n + F(b-a)(\frac{b-a}{n})^p f^{(q)}(\xi), \\ &= I_n + F\frac{(b-a)^{p+1}}{n^p} f^{(q)}(\xi). \end{aligned}$$

Sea $m = 2n$,

$$I = I_m + F\frac{(b-a)^{p+1}}{n^p 2^p} f^{(q)}(\zeta),$$

Supongamos que

$$f^{(q)}(\xi) \approx f^{(q)}(\zeta).$$

Entonces

$$I \approx I_n + 2^p G \approx I_n + e_n,$$
$$I \approx I_m + G \approx I_n + e_m,$$

donde $G = F\frac{(b-a)^{p+1}}{n^p 2^p} f^{(q)}(\zeta)$, e_n y e_m son los errores. Se puede despejar G:

$$\begin{aligned} e_m \approx G &= \frac{I_m - I_n}{2^p - 1} \\ &= \frac{I_m - I_n}{3} \qquad \text{trapecio} \\ &= \frac{I_m - I_n}{15} \qquad \text{Simpson} \end{aligned} \tag{6.10}$$

Con G se obtiene, supuestamente, una mejor aproximación de I:

$$I \approx I_m + G. \tag{6.11}$$

Los datos para el proceso iterativo para cuadratura adaptativa son: el método (la fórmula de Newton-Cotes), f, a, b, n_0, ε, $n_{\max}$.

Se empieza con un $n = n_0$ (debe ser adecuado) y se obtiene I_n. A partir de ahí se empieza a duplicar el número de subintervalos. El cálculo de la nueva aproximación I_m se hace **sin repetir evaluaciones de la función** f, ya que al duplicar el número de subintervalos los valores $f(x_i)$ de la etapa anterior hacen parte de los valores $f(x_j)$ de la etapa actual. Se calcula G aproximación de e_m, usando (6.10). Si $|G| \leq \varepsilon$, entonces se supone que el error es suficientemente pequeño y se toma como valor final $I_m + G$. En caso contrario, se continua duplicando el número de subintervalos. De todas está previsto un múmero máximo de subintervalos $n_{\max}$, ya que es posible que no se obtenga una aproximación del error suficientemente pequeña.

Ejemplo 6.5.

$$I = \int_0^\pi \operatorname{sen}(x)dx,$$

utilizando el método del trapecio ($n_0 = 1$) y el de Simpson, ($n_0 = 2$), $\varepsilon = 10^{-8}$

Método del trapecio:

n	I_n	G
1	0.0000000000000002	
2	1.5707963267948966	0.5235987755982988
4	1.8961188979370398	0.1084408570473811
8	1.9742316019455508	0.0260375680028370
16	1.9935703437723395	0.0064462472755962
32	1.9983933609701441	0.0016076723992682
64	1.9995983886400375	0.0004016758899645
128	1.9998996001842038	0.0001004038480554
256	1.9999749002350531	0.0000251000169498
512	1.9999937250705768	0.0000062749451746
1024	1.9999984312683834	0.0000015687326022
2048	1.9999996078171378	0.0000003921829181
4096	1.9999999019542845	0.0000000980457155
8192	1.9999999754885744	0.0000000245114300
16384	1.9999999938721373	0.0000000061278543

$I \approx$ `1.9999999938721373 + 0.0000000061278543= 1.9999999999999916.`

Método de Simpson:

n	I_n	G
2	2.0943951023931953	
4	2.0045597549844207	-0.0059890231605850
8	2.0002691699483881	-0.0002860390024022
16	2.0000165910479355	-0.0000168385933635
32	2.0000010333694127	-0.0000010371785682
64	2.0000000645300013	-0.0000000645892941
128	2.0000000040322572	-0.0000000040331829

$I \approx$ `2.0000000040322572 - 0.0000000040331829 = 1.9999999999990743.`

6.7 Cuadratura de Gauss

En las diferentes fórmulas de Newton-Cotes, los valores x_i deben estar igualmente espaciados. Esto se presenta con frecuencia cuando se dispone de una

tabla de valores $(x_i, f(x_i))$. En la cuadratura de Gauss se calcula la integral en un intervalo fijo $[-1, 1]$ mediante valores precisos pero no igualmente espaciados. Es decir, no se debe disponer de una tabla de valores, sino que debe ser posible evaluar la función en valores específicos.

La fórmula de cuadratura de Gauss tiene la forma

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} w_i\, f(x_i). \tag{6.12}$$

Los valores w_i se llaman los pesos o ponderaciones y los x_i son las abscisas. Si se desea integrar en otro intervalo,

$$\int_a^b \varphi(\xi)\,d\xi$$

es necesario hacer un cambio de variable,

$$t = \frac{2}{b-a}(\xi - a) - 1\,, \quad \xi = \frac{b-a}{2}(t+1) + a\,, \quad d\xi = \frac{b-a}{2}dt$$

$$\int_a^b \varphi(\xi)\,d\xi = \frac{b-a}{2}\int_{-1}^{1} \varphi(\frac{b-a}{2}(t+1)+a)\,dt,$$

$$\int_a^b \varphi(\xi)\,d\xi \approx \frac{b-a}{2}\sum_{i=1}^{n} w_i\,\varphi(\frac{b-a}{2}(x_i+1)+a), \tag{6.13}$$

$$\int_a^b \varphi(\xi)\,d\xi \approx \frac{b-a}{2}\sum_{i=1}^{n} w_i\,\varphi(\xi_i), \tag{6.14}$$

$$\xi_i = \frac{b-a}{2}(x_i+1) + a. \tag{6.15}$$

En la cuadratura de Gauss se desea que la fórmula (6.12) sea exacta para los polinomios de grado menor o igual que $m = m_n$, y se desea que este valor m_n sea lo más grande posible. En particular,

$$\int_{-1}^{1} f(x)\,dx = \sum_{i=1}^{n} w_i\, f(x_i)\,, \quad \text{si} \quad f(x) = 1, x, x^2, ..., x^{m_n}.$$

La anterior igualdad da lugar a $m_n + 1$ ecuaciones con $2n$ incógnitas (los w_i y los x_i). De donde $m_n = 2n - 1$, es decir, la fórmula (6.12) debe ser exacta para polinomios de grado menor o igual a $2n - 1$.

Recordemos que

$$\int_{-1}^{1} x^k \, dx = \begin{cases} 0 & \text{si } k \text{ es impar,} \\ \dfrac{2}{k+1} & \text{si } k \text{ es par.} \end{cases}$$

Para $n = 1$, se debe cumplir

$$w_1 = \int_{-1}^{1} 1 \, dx = 2,$$
$$w_1 x_1 = \int_{-1}^{1} x \, dx = 0.$$

Se deduce inmediatamente que

$$w_1 = 2\,, \quad x_1 = 0. \tag{6.16}$$

Para $n \geq 2$, se puede suponer, sin perder generalidad, que hay simetría en los valores x_i y en los pesos w_i. Más específicamente, se puede suponer que:

$$x_1 < x_2 < ... < x_n\,,$$
$$x_i = -x_{n+1-i}\,,$$
$$w_i = w_{n+1-i}\,.$$

Para $n = 2$,

$$w_1 + w_2 = \int_{-1}^{1} 1 \, dx = 2,$$
$$w_1 x_1 + w_2 x_2 = \int_{-1}^{1} x \, dx = 0,$$
$$w_1 x_1^2 + w_2 x_2^2 = \int_{-1}^{1} x^2 \, dx = \frac{2}{3},$$
$$w_1 x_1^3 + w_2 x_2^3 = \int_{-1}^{1} x^3 \, dx = 0.$$

Por suposiciones de simetría,

$$x_1 < 0 < x_2\,,$$
$$x_1 = -x_2\,,$$
$$w_1 = w_2.$$

Entonces

$$\begin{aligned} 2w_1 &= 2, \\ 2w_1x_1^2 &= \frac{2}{3}. \end{aligned}$$

Finalmente,

$$\begin{aligned} w_1 = 1, & x_1 = -\sqrt{\frac{1}{3}}, \\ w_2 = 1, & x_2 = \sqrt{\frac{1}{3}}. \end{aligned}$$

Para $n = 3$,

$$\begin{aligned} w_1 + w_2 + w_3 &= 2, \\ w_1x_1 + w_2x_2 + w_3x_3 &= 0, \\ w_1x_1^2 + w_2x_2^2 + w_3x_3^2 &= \frac{2}{3}, \\ w_1x_1^3 + w_2x_2^3 + w_3x_3^3 &= 0, \\ w_1x_1^4 + w_2x_2^4 + w_3x_3^4 &= \frac{2}{5}, \\ w_1x_1^5 + w_2x_2^5 + w_3x_3^5 &= 0. \end{aligned}$$

Por suposiciones de simetría,

$$\begin{aligned} & x_1 < 0 = x_2 < x_3\,, \\ & x_1 = -x_3\,, \\ & w_1 = w_3. \end{aligned}$$

Entonces

$$\begin{aligned} 2w_1 + w_2 &= 2, \\ 2w_1x_1^2 &= \frac{2}{3}, \\ 2w_1x_1^4 &= \frac{2}{5}. \end{aligned}$$

Finalmente,

$$w_1 = \frac{5}{9}, x_1 = -\sqrt{\frac{3}{5}},$$
$$w_2 = \frac{8}{9}, x_2 = 0,$$
$$w_3 = \frac{5}{9}, x_3 = \sqrt{\frac{3}{5}}.$$

La siguiente tabla contiene los valores w_i, x_i, para valores de n menores o iguales a 4.

n	w_i	x_i
1	2	0
2	1	$\pm$0.577350269189626
3	0.555555555555555	$\pm$0.774596669241483
	0.888888888888889	0
4	0.347854845137454	$\pm$0.861136311594053
	0.652145154862546	$\pm$0.339981043584856

Tablas más completas se pueden encontrar en [Fro70] o en [AbS74].

Ejemplo 6.6. Calcular una aproximación de

$$\int_{0.2}^{0.8} e^x \, dx$$

por cuadratura de Gauss con $n = 3$.

$$\xi_1 = \frac{0.8 - 0.2}{2}(-0.774596669241483 + 1) + 0.2 = 0.26762099922756$$
$$\xi_2 = \frac{0.8 - 0.2}{2}(0 + 1) + 0.2 = 0.5$$
$$\xi_3 = \frac{0.8 - 0.2}{2}(0.774596669241483 + 1) + 0.2 = 0.73237900077244$$

$$\int_{0.2}^{0.8} e^x \, dx \approx \frac{0.8 - 0.2}{2}\left(\frac{5}{9}\, e^{\xi_1} + \frac{8}{9}\, e^{\xi_2} + \frac{5}{9}\, e^{\xi_3}\right)$$
$$\approx 1.00413814737559$$

El valor exacto es $e^{0.8} - e^{0.2} = 1.00413817033230$, entonces el error es $0.00000002295671 \approx 2.3 \cdot 10^{-8}$. Si se emplea la fórmula de Simpson, que también utiliza tres evaluaciones de la función, se tiene

$$\int_{0.2}^{0.8} e^x \, dx \approx \frac{0.3}{3} \left(e^{0.2} + 4\, e^{0.5} + e^{0.8}\right) = 1.00418287694532$$

El error es $-0.00004470661302 \approx 4.5 \cdot 10^{-5}$. $\diamond$

La fórmula del error para 6.12 es:

$$e_n = \frac{2^{2n+1}(n!)^4}{(2n+1)((2n)!)^3} f^{(2n)}(\xi)\,, \quad -1 < \xi < 1\,. \tag{6.17}$$

Para 6.14 el error está dado por:

$$e_n = \frac{(b-a)^{2n+1}(n!)^4}{(2n+1)((2n)!)^3} f^{(2n)}(\xi)\,, \quad a < \xi < b\,. \tag{6.18}$$

Comparemos el método de Simpson y la fórmula de cuadratura de Gauss con $n = 3$, para integrar en el intervalo $[a, b]$, con $h = (b-a)/2$. En los dos casos es necesario evaluar tres veces la función.

$$\begin{aligned} e_{\text{Simpson}} &= -\frac{h^5}{90} f^{(4)}(z)\,, \\ e_{\text{Gauss3}} &= \frac{(2h)^7(3!)^4}{7(6!)^3} f^{(6)}(\xi) \;=\; \frac{h^7}{15750} f^{(6)}(\xi). \end{aligned}$$

Se observa que mientras que la fórmula de Simpson es exacta para polinomios de grado menor o igual a 3, la fórmula de Gauss es exacta hasta para polinomios de grado 5. Sea $0 < h < 1$. No sólo $h^7 < h^5$, sino que el coeficiente $1/15750$ es mucho menor que $1/90$.

En el ejemplo anterior, $h = 0.3$, y tanto $f^{(4)}$ como $f^{(6)}$ varían en el intervalo $[1.22,\ 2.23\ ]$.

$$\begin{aligned} e_{\text{Simpson}} &= -2.7 \cdot 10^{-5}\, f^{(4)}(z)\,, \\ e_{\text{Gauss3}} &= 1.39 \cdot 10^{-8} f^{(6)}(\xi)\,. \end{aligned}$$

6.7.1 Polinomios de Legendre

Las fórmulas de cuadratura vistas son las fórmulas de Gauss-Legendre. En ellas están involucrados los polinomios ortogonales de Legendre. También hay cuadratura de Gauss-Laguerre, de Gauss-Hermite y de Gauss-Chebyshev, relacionadas con los polinomios de Laguerre, de Hermite y de Chebyshev.

Hay varias maneras de definir los polinomios de Legendre; una de ellas es:

$$P_0(x) = 1, \tag{6.19}$$

$$P_n(x) = \frac{1}{2^n\, n!}\, \frac{\mathrm{d}^n}{\mathrm{d}x^n}(x^2-1)^n\,. \tag{6.20}$$

Por ejemplo,

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= x, \\ P_2(x) &= \frac{1}{2}(3x^2-1), \\ P_3(x) &= \frac{1}{2}(5x^3-x), \\ P_4(x) &= \frac{1}{8}(35x^4-30x^2+3). \end{aligned}$$

También existe una expresión recursiva:

$$P_0(x) = 1, \tag{6.21}$$

$$P_1(x) = x, \tag{6.22}$$

$$P_{n+1}(x) = \frac{2n+1}{n+1}\, x\, P_n(x) - \frac{n}{n+1} P_{n-1}(x)\,. \tag{6.23}$$

Algunas de las propiedades de los polinomios de Legendre son:

- $$\int_{-1}^{1} x^k P_n(x)\, dx = 0\,, \quad k = 0, 1, 2, ..., n-1, \tag{6.24}$$
- $$\int_{-1}^{1} P_m(x) P_n(x)\, dx = 0\,, \quad m \neq n, \tag{6.25}$$
- $$\int_{-1}^{1} (P_n(x))^2\, dx = \frac{2}{2n+1}\,\cdot \tag{6.26}$$

Las abscisas de las fórmulas de cuadratura de Gauss-Legendre son exactamente las raíces de $P_n(x)$. Además,

$$\bullet \quad w_i = \frac{1}{P_n'(x_i)} \int_{-1}^{1} \frac{P_n(x)}{x - x_i}\, dx, \tag{6.27}$$

$$\bullet \quad w_i = \frac{1}{(P_n'(x_i))^2} \; \frac{2}{1 - x_i^2}\,. \tag{6.28}$$

6.8 Derivación numérica

Dados los puntos (x_0, y_0), (x_1, y_1), ..., (x_n, y_n) igualmente espaciados en x, o sea, $x_i = x_0 + ih$, se desea tener aproximaciones de $f'(x_i)$ y $f''(x_i)$.

Como se vio anteriormente (5.6),

$$f(x) = p_n(x) + (x - x_0)(x - x_1)\cdots(x - x_n) f^{(n+1)}(\xi)/(n+1)!.$$

Sea $\Phi(x) = (x - x_0)(x - x_1)\cdots(x - x_n)$. Como ξ depende de x, se puede considerar $F(x) = f^{(n+1)}(\xi(x))/(n+1)!$. Entonces

$$\begin{aligned} f(x) &= p_n(x) + \Phi(x)F(x) \\ f'(x) &= p_n'(x) + \Phi'(x)F(x) + \Phi(x)F'(x), \\ f'(x_i) &= p_n'(x_i) + \Phi'(x_i)F(x_i) + \Phi(x_i)F'(x_i), \\ f'(x_i) &= p_n'(x_i) + \Phi'(x_i)F(x_i). \end{aligned}$$

Para $n = 1$

$$p_1(x) = y_0 + \frac{(y_1 - y_0)}{h}(x - x_0)\,, \quad p_1'(x) = \frac{(y_1 - y_0)}{h}.$$

$$\Phi(x) = (x - x_0)(x - x_1)\;, \quad \Phi'(x) = 2x - 2x_0 - h$$

Entonces

$$f'(x_0) = \frac{(y_1 - y_0)}{h} + (2x_0 - 2x_0 - h)F(x_0) = \frac{(y_1 - y_0)}{h} - \frac{h}{2} f''(\xi(x_0)),$$

$$f'(x_1) = \frac{(y_1 - y_0)}{h} + (2x_1 - 2x_0 - h)F(x_1) = \frac{(y_1 - y_0)}{h} + \frac{h}{2} f''(\xi(x_1)).$$

En general,

$$f'(x_i) = \frac{(y_{i+1} - y_i)}{h} \; - \frac{h}{2} f''(\xi), \;\; \xi \in [x_i, x_{i+1}] \tag{6.29}$$

$$f'(x_i) = \frac{(y_i - y_{i-1})}{h} \; + \frac{h}{2} f''(\zeta), \;\; \zeta \in [x_{i-1}, x_i] \tag{6.30}$$

El primer término después del signo igual corresponde al valor aproximado. El segundo término es el error. Se acostumbra decir que el error es del orden de h. Esto se escribe

$$f'(x_i) = \frac{(y_{i+1} - y_i)}{h} + O(h),$$
$$f'(x_i) = \frac{(y_i - y_{i-1})}{h} + O(h).$$

Para $n = 2$, sea $s = (x - x_0)/h$,

$$p_2(x) = y_0 + s\Delta f_0 + \frac{s(s-1)}{2}\frac{\Delta^2 f_0}{2},$$
$$p_2(x) = y_0 + \frac{x - x_0}{h}\Delta f_0 + \frac{x - x_0}{h}\ \frac{x - x_0 - h}{h}\ \frac{\Delta^2 f_0}{2},$$
$$p_2'(x) = \frac{\Delta f_0}{h} + \frac{2x - 2x_0 - h}{h^2}\ \frac{\Delta^2 f_0}{2},$$
$$p_2'(x_1) = \frac{\Delta f_0}{h} + \frac{\Delta^2 f_0}{2h} = \cdots$$
$$p_2'(x_1) = \frac{y_2 - y_0}{2h}.$$

$$\Phi(x) = (x - x_0)(x - x_0 - h)(x - x_0 - 2h),$$
$$\Phi(x) = (x - x_0)^3 - 3h(x - x_0)^2 + 2h^2(x - x_0),$$
$$\Phi'(x) = 3(x - x_0)^2 - 6h(x - x_0) + 2h^2,$$
$$\Phi'(x_1) = 3h^2 - 6h^2 + 2h^2 = -h^2.$$

Entonces

$$f'(x_1) = \frac{y_2 - y_0}{2h} - \frac{h^2}{6}f'''(\xi)\,, \quad \xi \in [x_0, x_2].$$

De manera general,

$$f'(x_i) = \frac{y_{i+1} - y_{i-1}}{2h} - \frac{h^2}{6}f'''(\xi)\,, \quad \xi \in [x_{i-1}, x_{i+1}], \tag{6.31}$$
$$f'(x_i) = \frac{y_{i+1} - y_{i-1}}{2h} + O(h^2).$$

En [YoG72], página 357, hay una tabla con varias fórmulas para diferenciación numérica. Para la segunda derivada, una fórmula muy empleada es:

$$f''(x_i) = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - \frac{h^2}{12}f^{(4)}(\xi)\,, \quad \xi \in [x_{i-1}, x_{i+1}], \tag{6.32}$$
$$f''(x_i) = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + O(h^2).$$

La deducción de las fórmulas de derivación numérica se hizo a partir de una tabla de valores (x_i, y_i), pero para el uso de éstas solamente se requiere conocer o poder evaluar f en los puntos necesarios. Por esta razón, algunas veces las fórmulas aparecen directamente en función de h:

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h), \tag{6.33}$$

$$f'(x) = \frac{f(x) - f(x-h)}{h} + O(h), \tag{6.34}$$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2), \tag{6.35}$$

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2). \tag{6.36}$$

Ejemplo 6.7. Dada $f(x) = \sqrt{x}$, evaluar aproximadamente $f'(4)$ y $f''(4)$, utilizando $h = 0.2$.

$$f'(4) \approx \frac{2.0494 - 2}{0.2} = 0.2470$$
$$f'(4) \approx \frac{2 - 1.9494}{0.2} = 0.2532$$
$$f'(4) \approx \frac{2.0494 - 1.9494}{2 \times 0.2} = 0.2501$$
$$f''(4) \approx \frac{2.0494 - 2 \times 2 + 1.9494}{0.2^2} = -0.0313\,. \ \diamond$$

El error de las dos primeras aproximaciones no es el mismo, pero es del mismo orden de magnitud $O(h)$. La tercera aproximación es mejor que las anteriores; su error es del orden de $O(h^2)$. Los valores exactos son $f'(4) = 0.25$, $f''(4) = -0.03125$.

6.8.1 Derivadas parciales

Sea $f : \mathbb{R}^n \to \mathbb{R}$ con derivadas dobles continuas. La fórmula (6.35) se puede generalizar a

$$\frac{\partial f}{\partial x_i}(\bar{x}) = \frac{1}{2h}\Big(f(\bar{x}_1, ..., \bar{x}_{i-1}, \bar{x}_i + h, \bar{x}_{i+1}, ..., \bar{x}_n) - f(\bar{x}_1, ..., \bar{x}_{i-1}, \bar{x}_i - h, \bar{x}_{i+1}, ..., \bar{x}_n)\Big) + O(h^2) \tag{6.37}$$

También se puede escribir de manera más compacta

$$\frac{\partial f}{\partial x_i}(\bar{x}) = \frac{f(\bar{x} + he^i) - f(\bar{x} - he^i)}{2h} + O(h^2) \tag{6.38}$$

donde

$$e^i = (0, ..., 0,1, 0, ..., 0) \in \mathbb{R}^n.$$

La fórmula (6.36) se puede generalizar a

$$\frac{\partial^2 f}{\partial x_i^2}(\bar{x}) = \frac{f(\bar{x} + he^i) - 2f(\bar{x}) + f(\bar{x} - he^i)}{h^2} + O(h^2) \tag{6.39}$$

Ejemplo 6.8. Sean $f(x_1, x_2) = e^{x_1} \operatorname{sen}(x_2)$. Obtenga una aproximación de $\dfrac{\partial f}{\partial x_2}(2,3)$ y de $\dfrac{\partial^2 f}{\partial x_1^2}(2,3)$ con $h = 0.2$.

$$\begin{aligned}
\frac{\partial f}{\partial x_2}(2,3) &\approx \frac{f(2,\ 3.2) - f(2,\ 2.8)}{0.4} \\
&= -7.2664401 \\
\frac{\partial^2 f}{\partial x_1^2}(2,3) &\approx \frac{f(2.2,\ 3) - 2f(2,\ 3) + f(1.8,\ 3)}{0.04} \\
&= 1.0462241
\end{aligned}$$

6.8.2 En Scilab

Sea $f : \mathbb{R} \to \mathbb{R}$ derivable. La aproximación de la derivada se obtiene por medio de `derivative(f, x)`. Si en un archivo se define la función

```
function y = func246(x)
  y = sqrt(x)
endfunction
```

y se carga este archivo en Scilab, entonces la derivada en $x = 4$ se obtiene mediante

```
der = derivative(func246, 4)
```

Si se quiere obtener también la segunda derivada:

```
[der, der2] = derivative(func246, 4)
```

Sea $f : \mathbb{R}^n \to \mathbb{R}$, por ejemplo, la definida en la siguiente función

```
function y = func245( x )
  y = exp(x(1)) * sin(x(2))
endfunction
```

Si se carga en Scilab el archivo donde está esta función, entonces para un vector columna `x`, la función `derivative` produce un vector fila con el gradiente.

```
x = [2 3]'
g = derivative(func245, x)
```

Para obtener, adicionalmente, la matriz hessiana:

```
x = [2 3]'
[g, A] = derivative(func245, x, H_form ='blockmat')
```

Sea $f : \mathbb{R}^n \to \mathbb{R}^m$, por ejemplo, la definida en la siguiente función

```
function fx = func247( x )
  fx = zeros(3,1)
  fx(1) = exp(x(1)) * sin(x(2))
  fx(2) = 3*x(1) + 4*x(2)
  fx(3) = x(1)*x(1) + 5*x(1)*x(2) + 3*x(2)*x(2)
endfunction
```

Si se carga en Scilab el archivo donde está esta función, entonces para un vector columna `x`, la función `derivative` produce una matriz $m \times n$, la matriz jacobiana.

```
x = [2 3]'
J = derivative(func247, x)
```

Ejercicios

6.1 Calcule

$$\int_{0.2}^{1} e^x \, dx$$

utilizando la fórmula del trapecio y de Simpson, variando el número de subintervalos. También por medio de la cuadratura de Gauss variando el número puntos. Calcule los errores. Compare.

6.2 Calcule

$$\int_{0}^{1} e^{-x^2} \, dx$$

utilizando la fórmula de Simpson. Utilice seis cifras decimales. Tome los valores $n = 2$, 4, 8, 16, 32... hasta que no haya variación.

6.3 Haga un programa para calcular $\int_a^b f(x)\,dx$, siguiendo el esquema del ejercicio anterior.

6.4 Observe, por ejemplo, que para $n = 2$ se evalúa la función en a, $(a + b)/2$, b. Para $n = 4$ se evalúa la función en a, $a + (b - a)/4$, $(a + b)/2$, $a + 3(b - a)/4$, b. Haga el programa eficiente para que no evalúe la función dos veces en el mismo punto.

6.5 Haga un programa para calcular $\int_a^b f(x)\,dx$, partiendo $[a, b]$ en subintervalos y utilizando en cada subintervalo cuadratura de Gauss.

6.6 Considere los puntos

$$\begin{gathered}(0.05, 2.0513),\\ (0.10, 2.1052),\\ (0.15, 2.1618),\\ (0.20, 2.2214),\\ (0.25, 2.2840),\\ (0.30, 2.3499),\\ (0.35, 2.4191),\\ (0.40, 2.4918).\end{gathered}$$

Calcule de la mejor manera posible

$$\int_{0.05}^{0.35} f(x)\,dx, \quad \int_{0.05}^{0.40} f(x)\,dx, \quad \int_{0.05}^{0.45} f(x)\,dx.$$

6.7 Considere los mismos puntos del ejercicio anterior. Calcule una aproximación de $f'(0.25)$, $f'(0.225)$, $f''(0.30)$.

6.8 Combine integración numérica y solución de ecuaciones para resolver

$$\int_0^x e^{-t^2}\,dt = 0.1.$$

7

Ecuaciones diferenciales

Este capítulo se refiere únicamente a ecuaciones diferenciales **ordinarias**. Generalmente una ecuación diferencial ordinaria de primer orden con condiciones iniciales, EDO1CI, se escribe de la forma

$$\begin{aligned} y' &= f(x,y) \text{ para } a \le x \le b, \\ y(x_0) &= y_0. \end{aligned} \tag{7.1}$$

Frecuentemente la condición inicial está dada sobre el extremo izquierdo del intervalo, o sea, $a = x_0$. Un ejemplo de EDO1CI es:

$$\begin{aligned} y' &= \frac{xy}{1+x^2+y^2} + 3x^2, \quad x \in [2,4], \\ y(2) &= 5. \end{aligned}$$

Temas importantísimos como existencia de la solución, unicidad o estabilidad, no serán tratados en este texto. El lector deberá remitirse a un libro de ecuaciones diferenciales. Aquí se supondrá que las funciones satisfacen todas las condiciones necesarias (continuidad, diferenciabilidad, condición de Lipschitz...) para que la solución exista, sea única...

Como en todos los otros casos de métodos numéricos, la primera opción para resolver una EDO1CI es buscar la solución analítica. Si esto no se logra, entonces se busca la solución numérica que consiste en encontrar valores aproximados y_1, y_2, ..., y_n tales que

$$y_i \approx y(x_i), \ i = 1, ..., n, \quad \text{donde } a = x_0 < x_1 < x_2 < ... < x_n = b.$$

En muchos casos los valores x_i están igualmente espaciados, o sea,

$$x_i = a + ih, \ i = 0, 1, ..., n, \quad \text{con } h = \frac{b-a}{n}.$$

En varios de los ejemplos siguientes se aplicarán los métodos numéricos para ecuaciones diferenciales con solución analítica conocida. Esto se hace simplemente para comparar la solución numérica con la solución exacta.

7.0.3 En Scilab

Consideremos la siguiente ecuación diferencial:

$$\begin{aligned} y' &= \frac{x+y}{x^2+y^2} + 4 + \cos(x)\,, \\ y(2) &= 3. \end{aligned}$$

Antes de utilizar la función `ode`, es necesario crear en Scilab la función f y cargarla. La función `ode` evalúa aproximaciones del valor de y en valores del tercer parámetro, un vector fila o columna o un número. El resultado es un vector fila con las aproximaciones de la solución en los valores deseados (tercer parámetro).

Despues de definir y cargar

```
function Dy = func158(x, y)
  Dy = ( x + y )/( x*x + y*y ) + 4 + cos(x)
endfunction
```

se obtiene la solución aproximada mediante

```
x0 = 2
y0 = 3
t = 2:0.05:3;
yt = ode(y0, x0, t, func158)
```

Ahora es posible graficar el resultado mediante

```
plot2d(t, yt)
```

7.1 Método de Euler

Se aplica a una EDO1CI como en (7.1) utilizando puntos igualmente espaciados. Su deducción es muy sencilla.

$$y'(x_0) \approx \frac{y(x_0+h)-y(x_0)}{h}.$$

Por otro lado

$$y'(x_0) = f(x_0, y_0).$$

Entonces

$$y(x_0+h) \approx y_0 + hf(x_0, y_0).$$

Si denotamos por y_1 la aproximación de $y(x_0+h)$, entonces la fórmula del método de Euler es justamente

$$y_1 = y_0 + hf(x_0, y_0).$$

Aplicando varias veces el mismo tipo de aproximaciones, se tiene la fórmula general:

$$y_{i+1} = y_i + hf(x_i, y_i). \tag{7.2}$$

Gráficamente esto significa que $y(x_i+h) = y(x_{i+1})$ se aproxima por el valor obtenido a partir de la recta tangente a la curva en el punto (x_i, y_i).

El valor y_1 es una aproximación de $y(x_1)$. A partir de y_1, no de $y(x_1)$, se hace una aproximación de $y'(x_1)$. Es decir, al suponer que y_2 es una aproximación de $y(x_2)$, se han hecho dos aproximaciones consecutivas y el error pudo haberse acumulado. De manera análoga, para decir que y_3 es una aproximación de $y(x_3)$, se han hecho tres aproximaciones, una sobre otra.

Sea $\varphi(t,h)$ definida para $t_1 \le t \le t_2$ y para valores pequeños de h. Se dice que

$$\varphi(t,h) = O(h^p)$$

si para valores pequeños de h existe una constante c tal que

$$|\varphi(t,h)| \le ch^p, \quad \forall t \in [t_1, t_2].$$

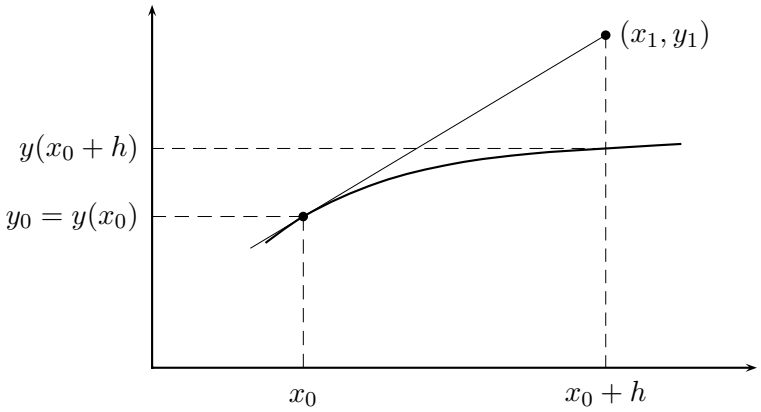


Figura 7.1: Método de Euler

También se acostumbra decir que

$$\varphi(t, h) \approx ch^p.$$

El error local tiene que ver con el error cometido para calcular $y(x_{i+1})$ suponiendo que y_i es un valor exacto, es decir, $y_i = y(x_i)$. El error global es el error que hay al considerar y_n como aproximación de $y(x_n)$ (n indica el número de intervalos).

Los resultados sobre el error en el método de Euler son:

$$y_1 = y(x_1) + O(h^2) \tag{7.3}$$
$$y_n = y(x_n) + O(h). \tag{7.4}$$

Ejemplo 7.1. Resolver, por el método de Euler, la ecuación diferencial

$$y' = 2x^2 - 4x + y$$
$$y(1) = 0.7182818$$

en el intervalo $[1, 3]$, con $h = 0.25$.

La primera observación es que esta ecuación diferencial se puede resolver analíticamente. Su solución es $y = e^x - 2x^2$. Luego no debería ser resuelta numéricamente. Sin embargo, el hecho de conocer su solución exacta permite

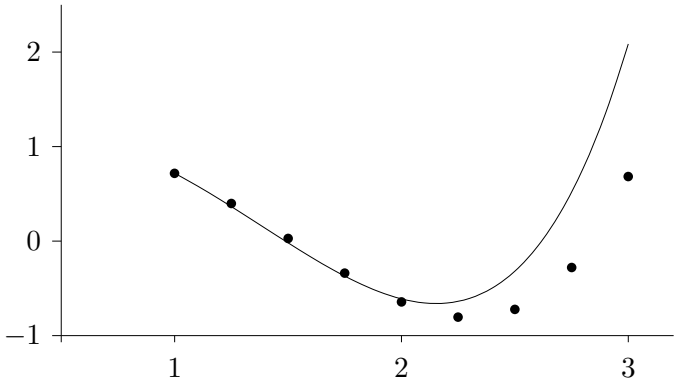


Figura 7.2: Ejemplo del método de Euler

ver el error cometido por el método numérico.

$$\begin{aligned}
y_1 &= y_0 + hf(x_0, y_0) \\
&= 0.7182818 + 0.25f(1,\, 0.7182818) \\
&= 0.7182818 + 0.25(0.7182818 + 2 \times 1^2 - 4 \times 1) \\
&= 0.3978523 \\
y_2 &= y_1 + hf(x_1, y_1) \\
&= 0.3978523 + 0.25f(1.25,\, 0.3978523) \\
&= 0.3978523 + 0.25(0.3978523 + 2 \times 1.25^2 - 4 \times 1.25) \\
&= 0.0285654 \\
y_3 &= ...
\end{aligned}$$

x_i	$\tilde{y}(x_i)$	$y(x_i)$
1.00	0.7182818	0.7182818
1.25	0.3978523	0.3653430
1.50	0.0285654	-0.0183109
1.75	-0.3392933	-0.3703973
2.00	-0.6428666	-0.6109439
2.25	-0.8035833	-0.6372642
2.50	-0.7232291	-0.3175060
2.75	-0.2790364	0.5176319
3.00	0.6824545	2.0855369

En los primeros valores se observa que el error es muy pequeño. A partir de $x = 2$ se empiezan a distanciar los valores $\tilde{y}(x)$ y $y(x)$. Si se trabaja con $h =$

0.1 se obtiene $\tilde{y}(3) = 1.4327409$; con $h = 0.01$ se obtiene $\tilde{y}(3) = 2.0133187$; con $h = 0.001$ se obtiene $\tilde{y}(3) = 2.0782381$. $\diamond$

El método de Euler se puede escribir en Scilab mediante:

```
function [Y, X] = Euler(f, x0, y0, xf, n)
  // Metodo de Euler para la ecuacion diferencial
  //
  // y' = f(x,y)
  // y(x0) = y0
  // en intervalo  [x0, xf]
  //
  // n = numero de subintervalos
  // Y, X seran vectores fila de  n+1 elementos
  // Y  contendra las aproximaciones de
  // y(x0) y(x0+h) y(x0+2h)  ... y(xf)
  // con h = (xf-x0)/n
  // X contendra los valores  x0  x0+h  x0+2h ... xf

  h = (xf-x0)/n
  X = zeros(1,n+1)
  Y = X
  X(1) = x0
  Y(1) = y0
  xi = x0
  yi = y0
  for i=1:n
    yi = yi + h*f(xi,yi)
    xi = xi+h
    Y(i+1) = yi
    X(i+1) = xi
  end
endfunction
```

7.2 Método de Heun

Este método es una modificación o mejora del método de Euler y se utiliza para el mismo tipo de problemas. También se conoce con el nombre de

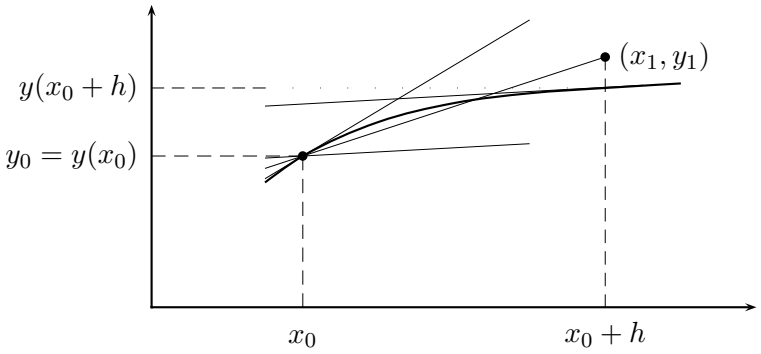


Figura 7.3: Método de Heun

método del trapecio. En el método de Euler se utiliza la aproximación

$$y(x + h) = y(x) + hy'(x).$$

En el método de Heun se busca cambiar, en la aproximación anterior, la derivada en x por un promedio de la derivada en x y en $x + h$.

$$y(x + h) \approx y(x) + h\,\frac{y'(x) + y'(x + h)}{2}$$

o sea,

$$y(x + h) \approx y(x) + h\,\frac{f(x, y(x)) + f(x + h, y(x + h))}{2}.$$

La fórmula anterior no se puede aplicar. Sirve para aproximar $y(x+h)$ pero utiliza $y(x + h)$. Entonces, en el lado derecho, se reemplaza $y(x + h)$ por la aproximación dada por el método de Euler

$$y(x + h) \approx y(x) + h\,\frac{f(x, y(x)) + f(x + h, y(x) + hf(x, y(x)))}{2}.$$

La anterior aproximación suele escribirse de la siguiente manera:

$$\begin{aligned} K_1 &= hf(x_i, y_i) \\ K_2 &= hf(x_i + h, y_i + K_1) \\ y_{i+1} &= y_i + \frac{1}{2}(K_1 + K_2). \end{aligned} \tag{7.5}$$

Ejemplo 7.2. Resolver, por el método de Heun, la ecuación diferencial

$$\begin{aligned} y' &= 2x^2 - 4x + y \\ y(1) &= 0.7182818 \end{aligned}$$

en el intervalo $[1, 3]$, con $h = 0.25$.

$$\begin{aligned} K_1 &= hf(x_0, y_0) \\ &= 0.25f(1,\ 0.7182818) \\ &= -0.320430 \\ K_2 &= hf(x_0 + h, y_0 + K_1) \\ &= 0.25f(1.25, 0.397852) \\ &= -0.369287 \\ y_1 &= y_0 + (K_1 + K_2)/2 \\ &= 0.3734236 \end{aligned}$$

$$\begin{aligned} K_1 &= hf(x_1, y_1) \\ &= 0.25f(1.25,\ 0.3734236) \\ &= -0.375394 \\ K_2 &= hf(x_1 + h, y_1 + K_1) \\ &= 0.25f(1.500000,\ -0.001971) \\ &= -0.375493 \\ y_2 &= y_1 + (K_1 + K_2)/2 \\ &= -0.0020198 \end{aligned}$$

$$K_1 = ...$$

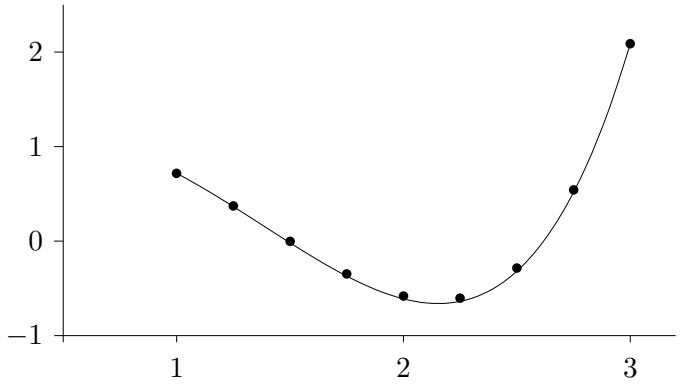


Figura 7.4: Ejemplo del método de Heun

x_i	$\tilde{y}(x_i)$	$y(x_i)$
1.00	0.7182818	0.7182818
1.25	0.3734236	0.3653430
1.50	-0.0020198	-0.0183109
1.75	-0.3463378	-0.3703973
2.00	-0.5804641	-0.6109439
2.25	-0.6030946	-0.6372642
2.50	-0.2844337	-0.3175060
2.75	0.5418193	0.5176319
3.00	2.0887372	2.0855369

En este ejemplo los resultados son mucho mejores. Por un lado, el método es mejor, pero, por otro, es natural tener mejores resultados pues hubo que evaluar 16 veces la función $f(x, y)$, 2 veces en cada iteración. En el ejemplo del método de Euler hubo simplemente 8 evaluaciones de la función $f(x, y)$. Al aplicar el método de Heun con $h = 0.5$ (es necesario evaluar 8 veces la función) se obtiene $\tilde{y}(3) = 2.1488885$, resultado no tan bueno como 2.0887372, pero netamente mejor que el obtenido por el método de Euler. Si se trabaja con $h = 0.1$ se obtiene $\tilde{y}(3) = 2.0841331$; con $h = 0.01$ se obtiene $\tilde{y}(3) = 2.0855081$; con $h = 0.001$ se obtiene $\tilde{y}(3) = 2.0855366$. $\diamond$

7.3 Método del punto medio

También este método es una modificación o mejora del método de Euler y se utiliza para el mismo tipo de problemas. En el método de Euler se utiliza

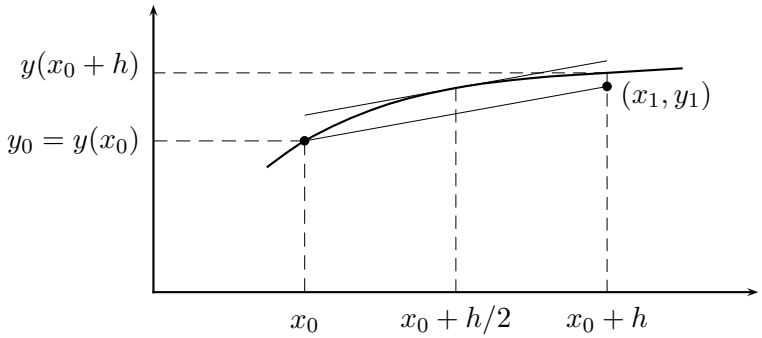


Figura 7.5: Método del punto medio

la aproximación

$$y(x+h) = y(x) + hy'(x).$$

En el método del punto medio se busca cambiar, en la aproximación anterior, la derivada en x por la derivada en el punto medio entre x y $x+h$, o sea, por la derivada en $x + h/2$.

$$y(x+h) \approx y(x) + h\, y'(x + h/2)$$

o sea,

$$y(x+h) \approx y(x) + h\, f(\, x + h/2, y(x+h/2)\,)\cdot$$

Como no se conoce $y(x+h/2)$, se reemplaza por la aproximación que daría el método de Euler con un paso de $h/2$.

$$y(x+h/2) \approx y(x) + \frac{h}{2} f(x,y)$$
$$y(x+h) \approx y(x) + h\, f(x + h/2, y(x) + \frac{h}{2} f(x,y))\cdot$$

La anterior aproximación suele escribirse de la siguiente manera:

$$\begin{aligned} K_1 &= hf(x_i, y_i) \\ K_2 &= hf(x_i + h/2, y_i + K_1/2) \\ y_{i+1} &= y_i + K_2. \end{aligned} \tag{7.6}$$

Ejemplo 7.3. Resolver, por el método del punto medio, la ecuación diferencial

$$\begin{aligned} y' &= 2x^2 - 4x + y \\ y(1) &= 0.7182818 \end{aligned}$$

en el intervalo $[1, 3]$, con $h = 0.25$.

$$\begin{aligned} K_1 &= hf(x_0, y_0) \\ &= 0.25f(1,\, 0.7182818) \\ &= -0.320430 \\ K_2 &= hf(x_0 + h/2, y_0 + K_1/2) \\ &= 0.25f(1.125, 0.558067) \\ &= -0.352671 \\ y_1 &= y_0 + K_2 \\ &= 0.3656111 \end{aligned}$$

$$\begin{aligned} K_1 &= hf(x_1, y_1) \\ &= 0.25f(1.25,\, 0.3656111) \\ &= -0.377347 \\ K_2 &= hf(x_1 + h/2, y_1 + K_1/2) \\ &= 0.25f(1.375,\, 0.176937) \\ &= -0.385453 \\ y_2 &= y_1 + K_2 \\ &= -0.0198420 \end{aligned}$$

$$K_1 = ...$$

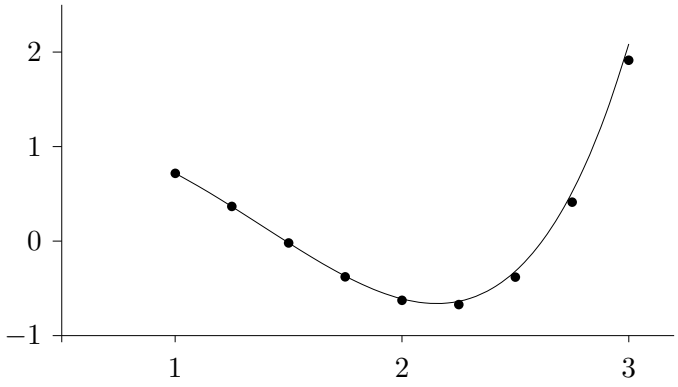


Figura 7.6: Ejemplo del método del punto medio

x_i	$\tilde{y}(x_i)$	$y(x_i)$
1.00	0.7182818	0.7182818
1.25	0.3656111	0.3653430
1.50	-0.0198420	-0.0183109
1.75	-0.3769851	-0.3703973
2.00	-0.6275434	-0.6109439
2.25	-0.6712275	-0.6372642
2.50	-0.3795415	-0.3175060
2.75	0.4121500	0.5176319
3.00	1.9147859	2.0855369

También, en este ejemplo, los resultados son mucho mejores. De nuevo hubo que evaluar 16 veces la función $f(x, y)$, 2 veces en cada iteración. Al aplicar el método del punto medio con $h = 0.5$ (es necesario evaluar 8 veces la función) se obtiene $\tilde{y}(3) = 1.5515985$, resultado no tan bueno como 2.0887372, pero netamente mejor que el obtenido por el método de Euler. Si se trabaja con $h = 0.1$ se obtiene $\tilde{y}(3) = 2.0538177$; con $h = 0.01$ se obtiene $\tilde{y}(3) = 2.0851903$; con $h = 0.001$ se obtiene $\tilde{y}(3) = 2.0855334$. $\diamond$

7.4 Método de Runge-Kutta

El método de Runge-Kutta o, más bien, los métodos de Runge-Kutta se aplican a una EDO1CI como en (7.1) utilizando puntos igualmente espaciados.

La forma general del método RK de orden n es la siguiente:

$$\begin{aligned} K_1 &= hf(x_i, y_i) \\ K_2 &= hf(x_i + \alpha_2 h,\ y_i + \beta_{21} K_1) \\ K_3 &= hf(x_i + \alpha_3 h,\ y_i + \beta_{31} K_1 + \beta_{32} K_2) \\ &\vdots \\ K_n &= hf(x_i + \alpha_n h,\ y_i + \beta_{n1} K_1 + \beta_{n2} K_2 + \cdots + \beta_{n,n-1} K_{n-1}) \\ y_{i+1} &= y_i + R_1 K_1 + R_2 K_2 + ... + R_n K_n. \end{aligned} \tag{7.7}$$

Se ve claramente que los métodos vistos son de RK: el método de Euler es uno de RK de orden 1, el método de Heun y el del punto medio son métodos de RK de orden 2.

Método de Euler:

$$\begin{aligned} K_1 &= hf(x_i, y_i) \\ y_{i+1} &= y_i + K_1. \end{aligned} \tag{7.8}$$

Método de Heun:

$$\begin{aligned} K_1 &= hf(x_i, y_i) \\ K_2 &= hf(x_i + h, y_i + K_1) \\ y_{i+1} &= y_i + \frac{1}{2} K_1 + \frac{1}{2} K_2. \end{aligned} \tag{7.9}$$

Método del punto medio:

$$\begin{aligned} K_1 &= hf(x_i, y_i) \\ K_2 &= hf(x_i + \frac{1}{2} h, y_i + \frac{1}{2} K_1) \\ y_{i+1} &= y_i + 0 K_1 + K_2. \end{aligned} \tag{7.10}$$

Un método muy popular es el siguiente método RK de orden 4:

$$\begin{aligned} K_1 &= hf(x_i, y_i) \\ K_2 &= hf(x_i + h/2,\ y_i + K_1/2) \\ K_3 &= hf(x_i + h/2,\ y_i + K_2/2) \\ K_4 &= hf(x_i + h,\ y_i + K_3) \\ y_{i+1} &= y_i + (K_1 + 2K_2 + 2K_3 + K_4)/6. \end{aligned} \tag{7.11}$$

Ejemplo 7.4. Resolver, por el método RK4 anterior, la ecuación diferencial

$$\begin{aligned} y' &= 2x^2 - 4x + y \\ y(1) &= 0.7182818 \end{aligned}$$

en el intervalo $[1, 3]$, con $h = 0.25$.

$$\begin{aligned} K_1 &= hf(x_0, y_0) \\ &= 0.25f(1,\ 0.7182818) \\ &= -0.320430 \\ K_2 &= hf(x_0 + h/2, y_0 + K_1/2) \\ &= 0.25f(1.125, 0.558067) \\ &= -0.352671 \\ K_3 &= hf(x_0 + h/2, y_0 + K_2/2) \\ &= 0.25f(1.125, 0.541946) \\ &= -0.356701 \\ K_4 &= hf(x_0 + h, y_0 + K_3) \\ &= 0.25f(1.25, 0.361581) \\ &= -0.378355 \\ y_1 &= y_0 + (K_1 + 2K_2 + 2K_3 + K_4)/6 \\ &= 0.3653606 \end{aligned}$$

$$\begin{aligned} K_1 &= hf(x_1, y_1) \\ &= 0.25f(1.25,\ 0.3653606) \\ &= -0.377410 \\ K_2 &= hf(x_1 + h/2, y_1 + K_1/2) \\ &= 0.25f(1.375, 0.176656) \\ &= -0.385524 \end{aligned}$$

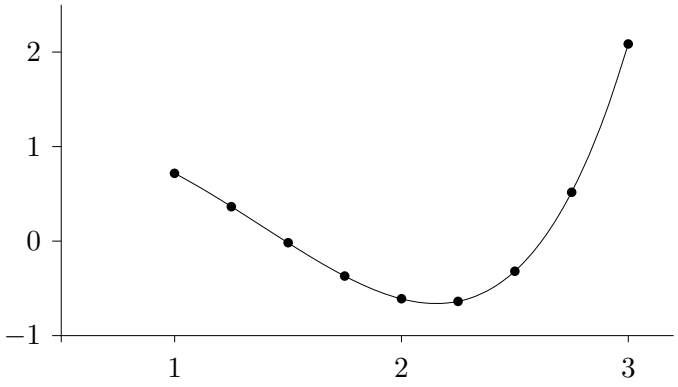


Figura 7.7: Ejemplo del método Runge-Kutta 4

$$\begin{aligned} K_3 &= hf(x_1 + h/2, y_1 + K_2/2) \\ &= 0.25f(1.375, 0.172599) \\ &= -0.386538 \\ K_4 &= hf(x_1 + h, y_1 + K_3) \\ &= 0.25f(1.5, -0.02117) \\ &= -0.380294 \\ y_2 &= y_1 + (K_1 + 2K_2 + 2K_3 + K_4)/6 \\ &= -0.0182773 \end{aligned}$$

x_i	$\tilde{y}(x_i)$	$y(x_i)$
1.00	0.7182818	0.7182818
1.25	0.3653606	0.3653430
1.50	-0.0182773	-0.0183109
1.75	-0.3703514	-0.3703973
2.00	-0.6108932	-0.6109439
2.25	-0.6372210	-0.6372642
2.50	-0.3174905	-0.3175060
2.75	0.5175891	0.5176319
3.00	2.0853898	2.0855369

En este ejemplo, los resultados son aún mejores. Hubo que evaluar 32 veces la función $f(x, y)$, 4 veces en cada iteración. Si se trabaja con $h = 0.1$ se obtiene $\tilde{y}(3) = 2.0855314$; con $h = 0.01$ se obtiene $\tilde{y}(3) = 2.0855369$; con $h = 0.001$ se obtiene $\tilde{y}(3) = 2.0855369$. $\diamond$

El método RK4 se puede escribir en Scilab de la siguiente manera:

```
function [Y, X] = RK4(f, x0, y0, xf, n)
  // Metodo Runge-Kutta 4 para la ecuacion diferencial
  //
  // y' = f(x,y)
  // y(x0) = y0
  // en intervalo  [x0, xf]
  //
  // n = numero de subintervalos
  //
  // Y, X seran vectores fila de  n+1 elementos
  // Y  contendra las aproximaciones de
  // y(x0) y(x0+h) y(x0+2h)  ... y(xf)
  // con h = (xf-x0)/n
  // X contendra los valores  x0  x0+h  x0+2h ... xf

  h = (xf-x0)/n
  X = zeros(1,n+1)
  Y = X
  X(1) = x0
  Y(1) = y0
  xi = x0
  yi = y0
  for i=1:n
    K1 = h*f(xi,     yi)
    K2 = h*f(xi+h/2, yi+K1/2);
    K3 = h*f(xi+h/2, yi+K2/2);
    K4 = h*f(xi+h,   yi+K3);
    xi = xi+h
    yi = yi + (K1 + 2*K2 + 2*K3 + K4)/6
    Y(i+1) = yi
    X(i+1) = xi
  end
endfunction
```

7.5 Deducción de RK2

En secciones anteriores se hizo la deducción, de manera más o menos intuitiva, de los métodos de Heun y del punto medio. Los dos resultan ser métodos de RK de orden 2. En esta sección veremos una deducción diferente y general de RK2.

El método RK2 tiene el siguiente esquema:

$$\begin{aligned} K_1 &= hf(x_i, y_i) \\ K_2 &= hf(x_i + \alpha_2 h,\ y_i + \beta_{21} K_1) \\ y_{i+1} &= y_i + R_1 K_1 + R_2 K_2. \end{aligned}$$

Como hay un solo coeficiente α y un solo coeficiente β, utilicémoslos sin subíndices:

$$\begin{aligned} K_1 &= hf(x_i, y_i) \\ K_2 &= hf(x_i + \alpha h,\ y_i + \beta K_1) \\ y_{i+1} &= y_i + R_1 K_1 + R_2 K_2. \end{aligned}$$

Sea g una función de dos variables. Si g es diferenciable en el punto $(\bar{u}, \bar{v})$, entonces se puede utilizar la siguiente aproximación de primer orden:

$$g(\bar{u} + \Delta u, \bar{v} + \Delta v) \approx g(\bar{u}, \bar{v}) + \Delta u \frac{\partial g}{\partial u}(\bar{u}, \bar{v}) + \Delta v \frac{\partial g}{\partial v}(\bar{u}, \bar{v}). \tag{7.12}$$

La aproximación de segundo orden para $y(x_i + h)$ es:

$$y(x_i + h) = y(x_i) + hy'(x_i) + \frac{h^2}{2} y''(x_i) + O(h^3) \tag{7.13}$$

$$y(x_i + h) \approx y(x_i) + hy'(x_i) + \frac{h^2}{2} y''(x_i). \tag{7.14}$$

En la aproximación anterior, podemos tener en cuenta que $y(x_i) = y_i$, y que $y'(x_i) = f(x_i, y_i)$. Además,

$$\begin{aligned} y''(x_i) &= \frac{d}{dx} y'(x_i) \\ &= \frac{d}{dx} f(x_i, y_i) \\ &= \frac{\partial f}{\partial x} f(x_i, y_i) + \frac{\partial f}{\partial y} f(x_i, y_i) \frac{\partial y}{\partial x}(x_i) \\ &= \frac{\partial f}{\partial x} f(x_i, y_i) + y'(x_i) \frac{\partial f}{\partial y} f(x_i, y_i). \end{aligned}$$

Para acortar la escritura utilizaremos la siguiente notación:

$$\begin{aligned} f &:= f(x_i, y_i) \\ f_x &:= \frac{\partial f}{\partial x} f(x_i, y_i) \\ f_y &:= \frac{\partial f}{\partial y} f(x_i, y_i) \\ y &:= y(x_i) \\ y' &:= y'(x_i) = f(x_i, y_i) = f \\ y'' &:= y''(x_i). \end{aligned}$$

Entonces

$$\begin{aligned} y'' &= f_x + f f_y \\ y(x_i + h) &\approx y + hf + \frac{h^2}{2} f_x + \frac{h^2}{2} f f_y. \end{aligned} \tag{7.15}$$

Por otro lado, el método RK2 se puede reescribir:

$$y_{i+1} = y_i + R_1 h f(x_i, y_i) + R_2 h f(x_i + \alpha h, y_i + \beta K_1).$$

Utilizando (7.12):

$$\begin{aligned} y_{i+1} = y_i &+ R_1 h f(x_i, y_i) \\ &+ R_2 h \left(f(x_i, y_i) + \alpha h \frac{\partial f}{\partial x}(x_i, y_i) + \beta K_1 \frac{\partial f}{\partial y}(x_i, y_i) \right). \end{aligned}$$

Utilizando la notación se obtiene:

$$\begin{aligned} y_{i+1} &= y + R_1 h f + R_2 h \left(f + \alpha h f_x + \beta K_1 f_y \right) \\ y_{i+1} &= y + (R_1 + R_2) h f + R_2 h^2 \alpha f_x + R_2 h \beta K_1 f_y. \end{aligned}$$

Como $K_1 = hf$, entonces

$$y_{i+1} = y + (R_1 + R_2) h f + R_2 \alpha h^2 f_x + R_2 \beta h^2 f f_y. \tag{7.16}$$

Al hacer la igualdad $y(x_i + h) = y_{i+1}$, en las ecuaciones (7.15) y (7.16) se comparan los coeficientes de hf, de $h^2 f_x$ y de $h^2 f f_y$ y se deduce:

$$\begin{aligned} R_1 + R_2 &= 1, \\ R_2 \alpha &= \frac{1}{2}, \\ R_2 \beta &= \frac{1}{2}. \end{aligned}$$

Entonces

$$\beta = \alpha, \tag{7.17}$$

$$R_2 = \frac{1}{2\alpha}\cdot \tag{7.18}$$

$$R_1 = 1 - R_2. \tag{7.19}$$

Si $\alpha = 1$, entonces $\beta = 1$, $R_2 = 1/2$ y $R_1 = 1/2$, es decir, el método de Heun. Si $\alpha = 1/2$, entonces $\beta = 1/2$, $R_2 = 1$ y $R_1 = 0$, es decir, el método del punto medio. Para otros valores de α se tienen otros métodos de RK de orden 2.

7.6 Control del paso

Hasta ahora se ha supuesto que para hallar la solución numérica de una ecuación diferencial, los puntos están igualmente espaciados, es decir, $x_i - x_{i-1} = h$ para $i = 1, 2, ..., n$. Esta política no es, en general, adecuada. Es preferible utilizar valores de h pequeños cuando es indispensable para mantener errores relativamente pequeños, y utilizar valores grandes de h cuando se puede.

Hay varios métodos para el control de h. En uno de ellos, se supone conocido y_i, una muy buena aproximación de $y(x_i)$, y se aplica un método con un paso h para obtener $\tilde{y}$ aproximación de $y(x_i + h)$. También se aplica el mismo método dos veces con el paso $h/2$ para obtener $\tilde{\tilde{y}}$, otra aproximación de $y(x_i + h)$. Con estos dos valores se puede acotar el error y así saber si es necesario trabajar con un paso más pequeño.

En otro enfoque, el que veremos en esta sección, se aplican dos métodos diferentes, con el mismo h y con estas dos aproximaciones se acota el error. Así se determina la buena o mala calidad de las aproximaciones.

Supongamos que tenemos dos métodos: el método A con error local $O(h^p)$ y el método B con error local $O(h^{p+1})$ (o con error local $O(h^q)$, $q \geq p + 1$). Partimos de y_i, muy buena aproximación de $y(x_i)$. Aplicando los dos métodos calculamos y_A y y_B, aproximaciones de $y(x_i + h)$. El control de paso tiene dos partes: en la primera se obtiene una aproximación del posible error obtenido.

$$|\text{error}| \approx e = \Phi_1(y_A, y_B, h, p).$$

Si e es menor o igual que un valor ε dado, entonces se acepta y_B como buena aproximación de $y(x+h)$. En caso contrario, es necesario utilizar un valor de h más pequeño. En ambos casos el valor de h se puede modificar, bien sea por necesidad ($e > \varepsilon$), bien sea porque, siendo h aceptable, es conveniente modificarlo para el siguiente paso. Para ello se calcula un coeficiente C_0 que sirve para obtener C coeficiente de h

$$\begin{aligned} C_0 &= \Phi_2(y_A, y_B, h, p) \\ C &= \varphi(C_0, ...) \\ h' &= Ch. \end{aligned}$$

Los diferentes algoritmos difieren en la manera de calcular e, C_0 y C (las funciones Φ_1, Φ_2 y φ). Más aún, para el mismo método A y el mismo método B hay diferentes algoritmos.

Un método muy popular es el de **Runge-Kutta-Fehlberg**, construido a partir de un método de RK de orden 5 (el método A) y de un método de RK de orden 6 (el método B). Una de sus ventajas está dada por el siguiente hecho: los valores K_1, K_2, K_3, K_4 y K_5 son los mismos para los dos métodos. Teniendo en cuenta la forma general (7.7) del método RK, basta con dar los valores α_i y β_{ij}. Recuérdese que siempre $K_1 = hf(x_i, y_i)$.

$$
\begin{array}{cccccc}
i & \alpha_i & \beta_{i1} & \beta_{i2} & \ldots & \\
2 & \frac{1}{4} & \frac{1}{4} & & & \\
3 & \frac{3}{8} & \frac{3}{32} & \frac{9}{32} & & \\
4 & \frac{12}{13} & \frac{1932}{2197} & -\frac{7200}{2197} & \frac{7296}{2197} & \\
5 & 1 & \frac{439}{216} & -8 & \frac{3680}{513} & -\frac{845}{4104} \\
6 & \frac{1}{2} & -\frac{8}{27} & 2 & -\frac{3544}{2565} & \frac{1859}{4104} \quad -\frac{11}{40}
\end{array}
\tag{7.20}
$$

$$
\begin{aligned}
y_A &= y_i + \frac{25}{216}K_1 + 0K_2 + \frac{1408}{2565}K_3 + \frac{2197}{4104}K_4 - \frac{1}{5}K_5 \\
y_B &= y_i + \frac{16}{135}K_1 + 0K_2 + \frac{6656}{12825}K_3 + \frac{28561}{56430}K_4 - \frac{9}{50}K_5 + \frac{2}{55}K_6
\end{aligned}
$$

Los errores locales son respectivamente $O(h^5)$ y $O(h^6)$. Realmente hay varias fórmulas RK5 y RK6; las anteriores están en [BuF85] y [EnU96]. Hay otras fórmulas diferentes en [ChC99].

La aproximación del error está dada por

$$
|\text{error}| \approx e = \frac{|y_A - y_B|}{h}. \tag{7.21}
$$

El coeficiente para la modificación del valor de h está dado por:

$$
\begin{aligned}
C_0 &= 0.84 \left(\frac{\varepsilon}{e}\right)^{1/4}, \\
C &= \min\{C_0, 4\}, \\
C &= \max\{C, 0.1\}.
\end{aligned}
\tag{7.22}
$$

Las fórmulas anteriores buscan que C no sea muy grande ni muy pequeño. Más específicamente, C debe estar en el intervalo $[0.1,\ 4]$.

En la descripción del algoritmo usaremos la siguiente notación de Matlab y de Scilab. La orden

```
u = [u; t]
```

significa que al vector columna `u` se le agrega al final el valor `t` y el resultado se llama de nuevo `u`.

MÉTODO RUNGE-KUTTA-FEHLBERG

datos: x_0, y_0, b, h_0, ε, h_{min}
$x = x_0$, $y = y_0$, $h = h_0$
$X = [x_0]$, $Y = [y_0]$
mientras $x < b$
  $h = \min\{h, b - x\}$
  `hbien` $= 0$
  mientras `hbien` $= 0$
    calcular y_a, y_B según (7.20)
    $e = |y_A - y_B|/h$
    si $e \leq \varepsilon$
      $x = x + h$, $y = y_B$
      `bienh` $= 1$
      $X = [X;\ x]$, $Y = [Y;\ y]$
    fin-si
    $C_0 = 0.84(\varepsilon/e)^{1/4}$
    $C = \max\{C_0, 0.1\}$, $C = \min\{C, 4\}$
    $h = Ch$
    si $h < h_{min}$ **ent parar**
  fin-mientras
fin-mientras

La salida no deseada del algoritmo anterior se produce cuando h se vuelve demasiado pequeño. Esto se produce en problemas muy difíciles cuando, para mantener el posible error dentro de lo establecido, ha sido necesario disminuir mucho el valor de h, por debajo del límite deseado.

En una versión ligeramente más eficiente, inicialmente no se calcula y_A ni y_B. Se calcula directamente

$$e = \left| \frac{1}{360}K_1 - \frac{128}{4275}K_3 - \frac{2197}{75240}K_4 + \frac{1}{50}K_5 + \frac{2}{55}K_6 \right|.$$

Cuando el valor de h es adecuado, entonces se calcula y_B para poder hacer la asignación $y = y_B$.

Ejemplo 7.5. Resolver, por el método RKF con control de paso, la ecuación

diferencial

$$\begin{aligned} y' &= 2x^2 - 4x + y \\ y(1) &= 0.7182818 \end{aligned}$$

en el intervalo $[1, 3]$, con $h_0 = 0.5$ y $\varepsilon = 10^{-6}$.

$$\begin{aligned} y_A &= -0.01834063 \\ y_B &= -0.01830704 \\ e &= 0.00006717 \end{aligned}$$

$h = 0.5$ no sirve.

$$\begin{aligned} C_0 &= 0.29341805 \\ C &= 0.29341805 \\ h &= 0.14670902 \\ y_A &= 0.51793321 \\ y_B &= 0.51793329 \\ e &= 0.00000057 \end{aligned}$$

$h = 0.14670902$ sirve.

$$\begin{aligned} x &= 1.14670902 \\ y &= 0.51793329 \\ C_0 &= 0.96535578 \\ C &= 0.96535578 \\ h &= 0.14162640 \\ y_A &= 0.30712817 \\ y_B &= 0.30712821 \\ e &= 0.00000029 \end{aligned}$$

$h = 0.14162640$ sirve.

$$\begin{aligned} x &= 1.28833543 \\ y &= 0.30712821 \\ &\vdots \end{aligned}$$

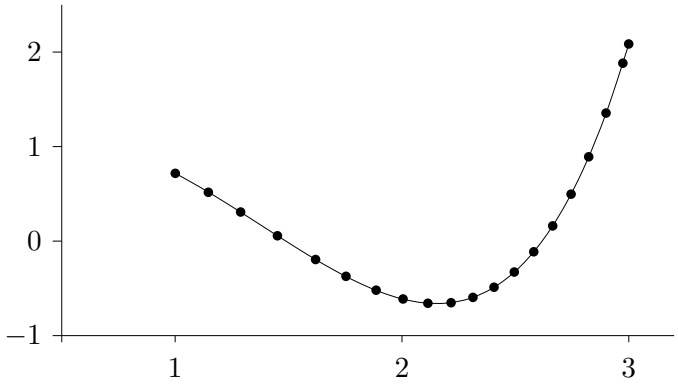


Figura 7.8: Ejemplo del método Runge-Kutta-Fehlberg

x	h	$\tilde{y}(x)$	$y(x)$
1.0000000	0.1467090	0.7182818	0.7182818
1.1467090	0.1416264	0.5179333	0.5179333
1.2883354	0.1622270	0.3071282	0.3071282
1.4505624	0.1686867	0.0572501	0.0572501
1.6192491	0.1333497	-0.1946380	-0.1946380
1.7525988	0.1329359	-0.3736279	-0.3736279
1.8855347	0.1191306	-0.5206051	-0.5206051
2.0046653	0.1092950	-0.6137572	-0.6137571
2.1139603	0.1024064	-0.6566848	-0.6566847
2.2163666	0.0971218	-0.6506243	-0.6506241
2.3134884	0.0928111	-0.5948276	-0.5948275
2.4062996	0.0891591	-0.4877186	-0.4877184
2.4954587	0.0859853	-0.3273334	-0.3273332
2.5814440	0.0831757	-0.1114979	-0.1114977
2.6646196	0.0806534	0.1620898	0.1620900
2.7452730	0.0783639	0.4958158	0.4958160
2.8236369	0.0762674	0.8921268	0.8921270
2.8999043	0.0743333	1.3535162	1.3535164
2.9742376	0.0257624	1.8825153	1.8825156
3.0000000		2.0855366	2.0855369

7.7 Orden del método y orden del error

Para algunos de los métodos hasta ahora vistos, todos son métodos de RK, se ha hablado del orden del método, del orden del error local y del orden del error global.

El orden del método se refiere al número de evaluaciones de la función f en cada iteración. Así por ejemplo, el método de Euler es un método de orden 1 y el método de Heun es un método de orden 2.

El orden del error local se refiere al exponente de h en el error teórico cometido en cada iteración. Si la fórmula es

$$y(x+h) = y(x) + R_1 k_1 + R_2 K_2 + \cdots + R_n K_n + O(h^p),$$

se dice que el error local es del orden de h^p, o simplemente, el error local es de orden p.

El orden del error global se refiere al exponente de h en el error obtenido al aproximar $y(b)$ después de hacer $(b - x_0)/h$ iteraciones.

Hemos visto seis métodos, Euler, Heun, punto medio, un RK4, un RK5 y un RK6. La siguiente tabla presenta los órdenes de los errores.

Método	Fórmula	Orden del método	Error local
Euler	(7.2)	1	$O(h^2)$
Heun	(7.5)	2	$O(h^3)$
Punto medio	(7.6)	2	$O(h^3)$
RK4	(7.11)	4	$O(h^5)$
RK5	(7.20)	5	$O(h^5)$
RK6	(7.20)	6	$O(h^6)$

El orden del error global es generalmente igual al orden del error local menos una unidad. Por ejemplo, el error global en el método de Euler es $O(h)$.

A medida que aumenta el orden del método, aumenta el orden del error, es decir, el error disminuye. Pero al pasar de RK4 a RK5 el orden del error no mejora. Por eso es más interesante usar el RK4 que el RK5 ya que se hacen solamente 4 evaluaciones y se tiene un error semejante. Ya con RK6 se obtiene un error más pequeño, pero a costa de dos evaluaciones más.

7.7.1 Verificación numérica del orden del error

Cuando se conoce la solución exacta de una ecuación diferencial, en muchos casos, se puede verificar el orden del error de un método específico. Más aún, se podría obtener el orden del error si éste no se conociera.

Sea $O(h^p)$ el error local del método. Se puede hacer la siguiente aproximación:

$$\text{error} = e \approx ch^p.$$

Al tomar logaritmo en la aproximación anterior se obtiene

$$\log(e) \approx \log(c) + p\log(h) \tag{7.23}$$

Para diferentes valores de h se evalúa el error cometido y se obtienen así varios puntos de la forma $(\log(h_i), \log(e_i))$. Estos puntos deben estar, aproximadamente, sobre una recta. La pendiente de esta recta es precisamente p. El valor de p se puede obtener gráficamente o por mínimos cuadrados.

Ejemplo 7.6. Obtener numéricamente el orden del error local del método de Heun usando la ecuación diferencial

$$y' = 2x^2 - 4x + y$$
$$y(1) = 0.7182818,$$

con $h = 0.1$, 0.12, 0.14, 0.16, 0.18 y 0.2.

h	$x_0 + h$	$\tilde{y}(x_0 + h)$	$y(x_0 + h)$	e	$\log(h)$	$\log(e)$
0.10	1.10	0.584701	0.584166	0.000535	-2.302585	-7.532503
0.12	1.12	0.556975	0.556054	0.000921	-2.120264	-6.989970
0.14	1.14	0.529024	0.527568	0.001456	-1.966113	-6.532007
0.16	1.16	0.500897	0.498733	0.002164	-1.832581	-6.135958
0.18	1.18	0.472641	0.469574	0.003067	-1.714798	-5.787212
0.20	1.20	0.444304	0.440117	0.004187	-1.609438	-5.475793

En la siguiente gráfica, $\log(h)$ en las abscisas y $\log(e)$ en las ordenadas, los puntos están aproximadamente en una recta.

Al calcular numéricamente los coeficientes de la recta de aproximación por mínimos cuadrados, se obtiene

$$\log(e) \approx 2.967325\ \log(h) - 0.698893$$
$$e \approx 0.497135h^{2.97}.$$

Estos resultados numéricos concuerdan con el resultado teórico. $\diamond$

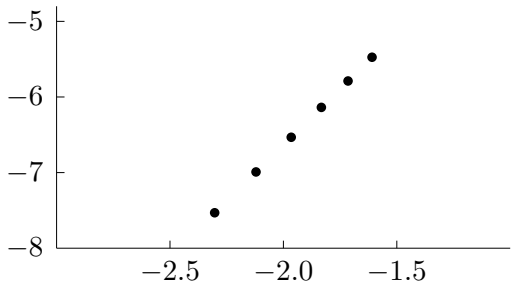


Figura 7.9: Orden local

7.8 Métodos multipaso explícitos

Los métodos RK son considerados como métodos monopaso (unipaso) por la siguiente razón. El valor y_{i+1} se calcula únicamente a partir del punto (x_i, y_i). En los métodos multipaso se utilizan otros puntos anteriores, por ejemplo, para calcular y_{i+1} se utilizan los puntos (x_{i-2}, y_{i-2}), (x_{i-1}, y_{i-1}) y (x_i, y_i).

Veamos un caso particular. Supongamos que se conocen los valores $y_0 = y(x_0)$, $y_1 = y(x_1)$ y $y_2 = y(x_2)$. Por facilidad para la deducción, supongamos que $x_0 = h$, $x_1 = h$ y $x_2 = 2h$.

Sea $p_2(x)$ el polinomio de grado menor o igual a 2 que interpola a f en los valores 0, h y $2h$, es decir, el polinomio pasa por los puntos $(0, f_0)$, (h, f_1) y $(2h, f_2)$, donde $f_i = f(x_i, y_i)$. Este polinomio se puede obtener utilizando polinomios de Lagrange:

$$p_2(x) = f_0\frac{(x-h)(x-2h)}{(0-h)(0-2h)} + f_1\frac{(x-0)(x-2h)}{(h-0)(h-2h)} + f_2\frac{(x-0)(x-h)}{(2h-0)(2h-h)}.$$

Después de algunas factorizaciones se obtiene:

$$p_2(x) = \frac{1}{2h^2}\left((f_0 - 2f_1 + f_2)x^2 + (-3f_0 + 4f_1 - f_2)hx + 2h^2 f_0\right).$$

Por otro lado, por el teorema fundamental del cálculo integral

$$\int_{x_2}^{x_3} y'(x)dx = y(x_3) - y(x_2)$$

$$y(x_3) = y(x_2) + \int_{x_2}^{x_3} y'(x)dx$$

$$y(x_3) = y(x_2) + \int_{2h}^{3h} f(x,y)dx.$$

Si se reemplaza $f(x,y)$ por el polinomio de interpolación, se tiene:

$$y(x_3) \approx y(x_2) + \int_{2h}^{3h} p_2(x)dx$$

$$y(x_3) \approx y(x_2) + \int_{2h}^{3h} \frac{1}{2h^2}\Big((f_0 - 2f_1 + f_2)x^2 + (-3f_0 + 4f_1 - f_2)hx + 2h^2 f_0\Big)dx$$

$$y_3 = y_2 + \frac{1}{2h^2}\Big((f_0 - 2f_1 + f_2)\frac{19}{3}h^3 + (-3f_0 + 4f_1 - f_2)\frac{5}{2}h^3 + 2h^3 f_0\Big)$$

$$y_3 = y_2 + \frac{h}{12}(5f_0 - 16f_1 + 23f_2) \tag{7.24}$$

La anterior igualdad se conoce con el nombre de fórmula de **Adams-Bashforth de orden 2** (se utiliza un polinomio de orden 2). También recibe el nombre de método multipaso explícito o método multipaso abierto de orden 2.

Si los valores y_0, y_1 y y_2 son exactos, o sea, si $y_0 = y(x_0)$, $y_1 = y(x_1)$ y $y_2 = y(x_2)$, entonces los valores f_i son exactos, o sea, $f(x_i, y_i) = f(x_i, y(x_i))$ y el error está dado por

$$y(x_3) = y(x_2) + \frac{h}{12}(5f_0 - 16f_1 + 23f_2) \;+\; \frac{3}{8}y^{(3)}(z)h^4, \quad z \in [x_0, x_3]. \tag{7.25}$$

La fórmula (7.24) se escribe en el caso general

$$y_{i+1} = y_i + \frac{h}{12}(5f_{i-2} - 16f_{i-1} + 23f_i). \tag{7.26}$$

Para empezar a aplicar esta fórmula se requiere conocer los valores f_j anteriores. Entonces es indispensable utilizar un método RK el número de veces necesario. El método RK escogido debe ser de mejor calidad que el método de Adams-Bashforth que estamos utilizando. Para nuestro caso podemos utilizar RK4.

Ejemplo 7.7. Resolver, por el método de Adams-Bashforth de orden 2, la ecuación diferencial

$$y' = 2x^2 - 4x + y$$
$$y(1) = 0.7182818$$

en el intervalo $[1, 3]$, con $h = 0.25$.

Al aplicar el método RK4 dos veces se obtiene:

$$\begin{aligned} y_1 &= \ \ 0.3653606 \\ y_2 &= -0.0182773. \end{aligned}$$

Entonces

$$\begin{aligned} f_0 &= f(x_0, y_0) = -1.2817182 \\ f_1 &= f(x_1, y_1) = -1.5096394 \\ f_2 &= -1.5182773 \\ y_3 &= y_2 + h(5f_0 - 16f_1 + 23f_2)/12 \\ &= -0.3760843 \\ f_3 &= f(x_3, y_3) = -1.2510843 \\ y_4 &= -0.6267238 \\ &\vdots \end{aligned}$$

x_i	$\tilde{y}(x_i)$	$y(x_i)$
1.00	0.7182818	0.7182818
1.25	0.3653606	0.3653430
1.50	-0.0182773	-0.0183109
1.75	-0.3760843	-0.3703973
2.00	-0.6267238	-0.6109439
2.25	-0.6681548	-0.6372642
2.50	-0.3706632	-0.3175060
2.75	0.4320786	0.5176319
3.00	1.9534879	2.0855369

En este caso hubo que evaluar 8 veces la función para los dos valores de RK4 y en seguida 6 evaluaciones para un total de 14 evaluaciones de la función f. $\diamond$

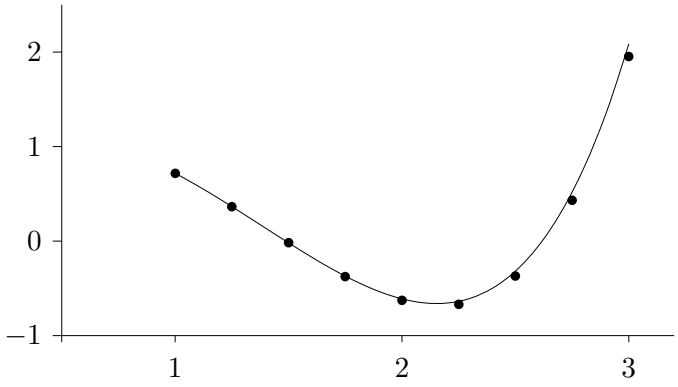


Figura 7.10: Ejemplo del método de Adams-Bashforth 2

MULTIPASO EXPLÍCITO: ADAMS-BASHFORTH

n		error
0	$y_{i+1} = y_i + hf_i$	$\frac{1}{2}y''(\xi)h^2$
1	$y_{i+1} = y_i + \frac{h}{2}(-f_{i-1} + 3f_i)$	$\frac{5}{12}y'''(\xi)h^3$
2	$y_{i+1} = y_i + \frac{h}{12}(5f_{i-2} - 16f_{i-1} + 23f_i)$	$\frac{3}{8}y^{(4)}(\xi)h^4$
3	$y_{i+1} = y_i + \frac{h}{24}(-9f_{i-3} + 37f_{i-2} - 59f_{i-1} + 55f_i)$	$\frac{251}{720}y^{(5)}(\xi)h^5$
4	$y_{i+1} = y_i + \frac{h}{720}(251f_{i-4} - 1274f_{i-3} + 2616f_{i-2} - 2774f_{i-1} + 1901f_i)$	$\frac{95}{288}y^{(6)}(\xi)h^6$

En la anterior tabla se muestran las principales fórmulas. Allí n indica el grado del polinomio de interpolación usado. En algunos libros, n está asociado con número de puntos utilizados para la interpolación (igual al grado del polinomio más uno). Obsérvese que la primera fórmula es simplemente el método de Euler.

7.9 Métodos multipaso implícitos

En estos métodos se utiliza un polinomio de interpolación, el mismo de los métodos explícitos, pero el intervalo de integración varía.

Veamos un caso particular. Supongamos que se conocen los valores $y_0 = y(x_0)$, $y_1 = y(x_1)$ y $y_2 = y(x_2)$. Por facilidad para la deducción, supongamos que $x_0 = h$, $x_1 = h$ y $x_2 = 2h$.

Sea $p_2(x)$ el polinomio de grado menor o igual a 2 que interpola a f en los valores 0, h y $2h$, es decir, el polinomio pasa por los puntos $(0, f_0)$, (h, f_1) y $(2h, f_2)$, donde $f_i = f(x_i, y_i)$. Como se vio en la sección anterior,

$$p_2(x) = \frac{1}{2h^2}\left((f_0 - 2f_1 + f_2)x^2 + (-3f_0 + 4f_1 - f_2)hx + 2h^2 f_0\right).$$

El teorema fundamental del cálculo integral se usa de la siguiente manera:

$$\int_{x_1}^{x_2} y'(x)dx = y(x_2) - y(x_1)$$
$$y(x_2) = y(x_1) + \int_{x_1}^{x_2} y'(x)dx$$
$$y(x_2) = y(x_1) + \int_{h}^{2h} f(x, y)dx.$$

Si se reemplaza $f(x, y)$ por el polinomio de interpolación se tiene:

$$y(x_2) \approx y(x_1) + \int_{h}^{2h} p_2(x)dx$$
$$y(x_2) \approx y(x_1) + \int_{h}^{2h} \frac{1}{2h^2}\Big((f_0 - 2f_1 + f_2)x^2 + (-3f_0 + 4f_1 - f_2)hx + 2h^2 f_0\Big)dx$$

$$y_2 = y_1 + \frac{1}{2h^2}\Bigg((f_0 - 2f_1 + f_2)\frac{7}{3}h^3 + (-3f_0 + 4f_1 - f_2)\frac{3}{2}h^3 + 2h^3 f_0\Bigg)$$

$$y_2 = y_1 + \frac{h}{12}(-f_0 + 8f_1 + 5f_2). \tag{7.27}$$

La anterior igualdad se conoce con el nombre de fórmula de **Adams-Moulton de orden 2** (se utiliza un polinomio de orden 2). También recibe el nombre de método multipaso implícito o método multipaso cerrado de orden 2.

Si los valores y_0, y_1 y y_2 son exactos, o sea, si $y_0 = y(x_0)$, $y_1 = y(x_1)$ y $y_2 = y(x_2)$, entonces los valores f_i son exactos, o sea, $f(x_i, y_i) = f(x_i, y(x_i))$ y el error está dado por

$$y(x_2) = y(x_1) + \frac{h}{12}(-f_0 + 8f_1 + 5f_2) \;-\; \frac{1}{24}y^{(3)}(z)h^4, \quad z \in [x_0, x_2]. \tag{7.28}$$

La fórmula (7.27) se escribe en el caso general

$$y_{i+1} = y_i + \frac{h}{12}(-f_{i-1} + 8f_i + 5f_{i+1}). \tag{7.29}$$

Para empezar a aplicar esta fórmula es indispensable conocer los valores f_j anteriores. Entonces se requiere utilizar un método RK el número de veces necesario. El método RK escogido debe ser de mejor calidad que el método de Adams-Bashforth que estamos utilizando. Para nuestro caso podemos utilizar RK4.

Una dificultad más grande, y específica de los métodos implícitos, está dada por el siguiente hecho: para calcular y_{i+1} se utiliza f_{i+1}, pero este valor es justamente $f(x_{i+1}, y_{i+1})$. ¿Cómo salir de este círculo vicioso? Inicialmente se calcula y^0_{i+1}, una primera aproximación, por el método de Euler. Con este valor se puede calcular $f^0_{i+1} = f(x_{i+1}, y^0_{i+1})$ y en seguida y^1_{i+1}. De nuevo se calcula $f^1_{i+1} = f(x_{i+1}, y^1_{i+1})$ y en seguida y^2_{i+1}. Este proceso iterativo acaba cuando dos valores consecutivos, y^k_{i+1} y y^{k+1}_{i+1}, son muy parecidos. Este método recibe también el nombre de método **predictor-corrector**. La fórmula queda entonces así:

$$y^{k+1}_{i+1} = y_i + \frac{h}{12}(-f_{i-1} + 8f_i + 5f^k_{i+1}). \tag{7.30}$$

El criterio de parada puede ser:

$$\frac{|y_i^{k+1} - y_i^k|}{\max\{1, |y_i^{k+1}|\}} \leq \varepsilon.$$

Ejemplo 7.8. Resolver, por el método de Adams-Moulton de orden 2, la ecuación diferencial

$$\begin{aligned} y' &= 2x^2 - 4x + y \\ y(1) &= 0.7182818 \end{aligned}$$

en el intervalo $[1, 3]$, con $h = 0.25$ y $\varepsilon = 0.0001$.

Al aplicar el método RK4 una vez, se obtiene:

$$y_1 = \ 0.3653606$$

Entonces

$$\begin{aligned} f_0 &= f(x_0, y_0) = -1.2817182 \\ f_1 &= f(x_1, y_1) = -1.5096394 \end{aligned}$$

Aplicando Euler se obtiene una primera aproximación de y_2:

$$\begin{aligned} y_2^0 &= -0.0120493 \\ f_2^0 &= -1.5120493 \end{aligned}$$

Empiezan las iteraciones:

$$\begin{aligned} y_2^1 &= -0.0170487 \\ f_2^1 &= -1.5170487 \\ y_2^2 &= -0.0175694 \\ f_2^2 &= -1.5175694 \\ y_2^3 &= -0.0176237 = y_2 \end{aligned}$$

Para calcular y_2 se utilizan los valores:

$$\begin{aligned} f_1 &= -1.5096394 \\ f_2 &= -1.5176237. \end{aligned}$$

Aplicando Euler se obtiene una primera aproximación de y_3:

$$y_3^0 = -0.3970296$$
$$f_3^0 = -1.2720296$$

Empiezan las iteraciones:

$$y_3^1 = -0.3716132$$
$$f_3^1 = -1.2466132$$
$$y_3^2 = -0.3689657$$
$$f_3^2 = -1.2439657$$
$$y_3^3 = -0.3686899$$
$$f_3^3 = -1.2436899$$
$$y_3^4 = -0.3686612 = y_3$$
$$\vdots$$

x_i	$\tilde{y}(x_i)$	$y(x_i)$
1.00	0.7182818	0.7182818
1.25	0.3653606	0.3653430
1.50	-0.0176237	-0.0183109
1.75	-0.3686612	-0.3703973
2.00	-0.6076225	-0.6109439
2.25	-0.6315876	-0.6372642
2.50	-0.3084043	-0.3175060
2.75	0.5316463	0.5176319
3.00	2.1065205	2.0855369

En este caso hubo que evaluar 4 veces la función para el valor de RK4 y en seguida, en cada uno de los otros 7 intervalos, una evaluación fija más las requeridas al iterar. En este ejemplo hubo, en promedio, 4 por intervalo, para un total de 32 evaluaciones de f. El valor final y_8 es más exacto que el obtenido por Adams-Bashforth, pero a costa de más evaluaciones. $\diamond$

Teóricamente, los dos métodos multipaso de orden 2 tienen un error local del mismo orden, $O(h^4)$, pero el coeficiente en el método multipaso explícito, 3/8, es nueve veces el coeficiente en el error del método implícito, 1/24.

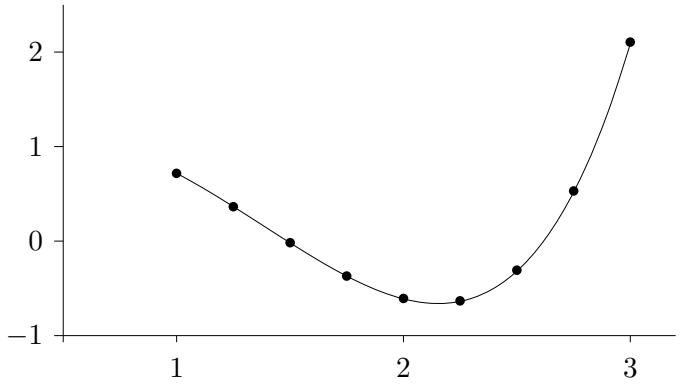


Figura 7.11: Ejemplo del método de Adams-Moulton 2

MULTIPASO IMPLÍCITO: ADAMS-MOULTON

n		error
1	$y_{i+1} = y_i + \frac{h}{2}(f_i + f_{i+1})$	$-\frac{1}{12}y''(\xi)h^3$
2	$y_{i+1} = y_i + \frac{h}{12}(-f_{i-1} + 8f_i + 5f_{i+1})$	$-\frac{1}{24}y^{(3)}(\xi)h^4$
3	$y_{i+1} = y_i + \frac{h}{24}(f_{i-2} - 5f_{i-1} + 19f_i + 9f_{i+1})$	$-\frac{19}{720}y^{(4)}(\xi)h^5$
4	$y_{i+1} = y_i + \frac{h}{720}(-19f_{i-3} + 106f_{i-2} - 264f_{i-1} + 646f_i + 251f_{i+1})$	$-\frac{27}{1440}y^{(5)}(\xi)h^6$

La tabla anterior contiene las principales fórmulas. Allí n indica el grado del polinomio de interpolación usado. Obsérvese que el método de Heun corresponde a una iteración (una sola) del método multipaso implícito de orden 1.

7.10 Sistemas de ecuaciones diferenciales

Un sistema de m ecuaciones diferenciales de primer orden se puede escribir de la siguiente forma:

$$\begin{aligned}
\frac{dy_1}{dx} &= f_1(x, y_1, y_2, ..., y_m) \\
\frac{dy_2}{dx} &= f_2(x, y_1, y_2, ..., y_m) \\
&\vdots \\
\frac{dy_m}{dx} &= f_m(x, y_1, y_2, ..., y_m)
\end{aligned}$$

para $x_0 \leq x \leq b$, con las condiciones iniciales

$$\begin{aligned}
y_1(x_0) &= y_1^0 \\
y_2(x_0) &= y_2^0 \\
&\vdots \\
y_m(x_0) &= y_m^0.
\end{aligned}$$

Utilicemos la siguiente notación:

$$\begin{aligned}
y &= (y_1, y_2, ..., y_m) \\
y^0 &= (y_1^0, y_2^0, ..., y_m^0) \\
f(x, y) &= f(x, y_1, y_2, ..., y_m) \\
&= (\, f_1(x, y_1, ..., y_m), f_2(x, y_1, ..., y_m), ..., f_m(x, y_1, ..., y_m)\,).
\end{aligned}$$

De esta manera, el sistema se puede escribir así:

$$\begin{aligned}
y' &= f(x, y), \quad x_0 \leq x \leq b \\
y(x_0) &= y^0.
\end{aligned}$$

La solución numérica del sistema de ecuaciones consiste en un conjunto de vectores y^0, y^1, y^2, ..., y^n,

$$y^i = (y_1^i, y_2^i, ..., y_m^i),$$

donde cada y_j^i es una aproximación:

$$y_j^i \approx y_j(x_k).$$

Los métodos vistos anteriormente se pueden generalizar de manera inmediata. Si se trata de los método RK, entonces los K_i dejan de ser números y pasan a ser vectores K^i. Para y se utiliza un superíndice para indicar el intervalo, ya que los subíndices se usan para las componentes del vector. Por ejemplo, las fórmulas de RK4 se convierten en:

$$\begin{aligned} K^1 &= hf(x_i, y^i) \\ K^2 &= hf(x_i + h/2,\ y^i + K^1/2) \\ K^3 &= hf(x_i + h/2,\ y^i + K^2/2) \\ K^4 &= hf(x_i + h,\ y^i + K^3) \\ y^{i+1} &= y^i + (K^1 + 2K^2 + 2K^3 + K^4)/6. \end{aligned} \tag{7.31}$$

Ejemplo 7.9. Resolver el siguiente sistema de ecuaciones por RK4:

$$\begin{aligned} y_1' &= \frac{2y_1}{x} + x^3 y_2, \quad 1 \le x \le 2 \\ y_2' &= -\frac{3}{x} y_2 \\ y_1(1) &= -1 \\ y_2(1) &= \ \ 1 \end{aligned}$$

con $h = 0.2$.

La solución (exacta) de este sencillo sistema de ecuaciones es:

$$\begin{aligned} y_1(x) &= -x \\ y_2(x) &= \ \ x^{-3}. \end{aligned}$$

Para la solución numérica:

$$
\begin{aligned}
K^1 &= (-0.2,\ -0.6)\\
K^2 &= (-0.2136600,\ -0.3818182)\\
K^3 &= (-0.1871036,\ -0.4413223)\\
K^4 &= (-0.2026222,\ -0.2793388)\\
y^1 &= (-1.2006916,\ 0.5790634)
\end{aligned}
$$

$$
\begin{aligned}
K^1 &= (-0.2001062,\ -0.2895317)\\
K^2 &= (-0.2093988,\ -0.2004450)\\
K^3 &= (-0.1912561,\ -0.2210035)\\
K^4 &= (-0.2011961,\ -0.1534542)\\
y^2 &= (-1.4011269,\ 0.3647495)\\
&\vdots
\end{aligned}
$$

x_i	$\tilde{y}_1(x_i)$	$\tilde{y}_2(x_i)$	$y_1(x_i)$	$y_2(x_i)$
1.0	-1.0	1.0	-1.0	1.0
1.2	-1.2006916	0.5790634	-1.2	0.5787037
1.4	-1.4011269	0.3647495	-1.4	0.3644315
1.6	-1.6014497	0.2443822	-1.6	0.2441406
1.8	-1.8017156	0.1716477	-1.8	0.1714678
2.0	-2.0019491	0.1251354	-2.0	0.125

$\diamond$

7.10.1 En Scilab

Consideremos el siguiente sistema de ecuaciones diferenciales:

$$
\begin{aligned}
y_1' &= \frac{2y_1}{x} + x^3 y_2,\\
y_2' &= -\frac{3}{x} y_2\\
y_1(1) &= -1\\
y_2(1) &= \ \ 1.
\end{aligned}
$$

Despues de definir y cargar

```
function fxy = func43(x, y)
  fxy = zeros(2,1)
  fxy(1) = 2*y(1)/x + x^3*y(2)
  fxy(2) = -3*y(2)/x
endfunction
```

se utiliza la misma función `ode`, pero con los parámetros de dimensión adecuada.

```
x0 = 1
y0 = [-1 1]'
t = (1:0.2:2)'
yt = ode(y0, x0, t, func43)
```

En este caso, `yt` es un matriz de dos filas. En la fila i están las aproximaciones de los valores de $y_i(t_j)$.

Escribir una función en Scilab para un sistema de ecuaciones diferenciales es casi igual a la función para una ecuación diferencial. A continuación una versión del método RK4 para sistemas.

```
function [Y, X] = RK4Sist(f, x0, y0, xf, n)
  // Metodo Runge-Kutta 4 para sistema de ecuaciones diferenciales
  //
  // y' = f(x,y)
  // y(x0) = y0
  // en intervalo  [x0, xf]
  //
  // x0  es un numero
  // y0  es un vector columna, digamos  p x 1.
  // La funcion  f  tiene dos parametros,
  // x  un numero,  y  un vector columna,
  // su resultado es un vector columna.

  // n = numero de subintervalos.
  //
  // Y sera una matriz con p filas, n+1 columnas.
  // X sera un  vector fila de  n+1 elementos.
  // Cada columna de Y  contendra las aproximaciones de
```

```
  // y(x0) y(x0+h) y(x0+2h)  ... y(xf)
  // con h = (xf-x0)/n
  // X contendra los valores  x0  x0+h  x0+2h ... xf

  h = (xf-x0)/n
  p = size(y0,1)
  disp(p, 'p')
  X = zeros(1,n+1)
  Y = zeros(p,n+1)
  X(1) = x0
  Y(:,1) = y0
  xi = x0
  yi = y0
  for i=1:n
    K1 = h*f(xi,     yi)
    K2 = h*f(xi+h/2, yi+K1/2);
    K3 = h*f(xi+h/2, yi+K2/2);
    K4 = h*f(xi+h,   yi+K3);
    xi = xi+h
    yi = yi + (K1 + 2*K2 + 2*K3 + K4)/6
    Y(:,i+1) = yi
    X(i+1) = xi
  end
endfunction
```

7.11 Ecuaciones diferenciales de orden superior

Una ecuación diferencial ordinaria, de orden m, con condiciones iniciales, se puede escribir de la siguiente manera:

$$\begin{aligned} y^{(m)} &= f(x, y, y', y'', ..., y^{(m-1)}), \quad x_0 \leq x \leq b \\ y(x_0) &= y_0 \\ y'(x_0) &= y_0' \\ y''(x_0) &= y_0'' \\ &\vdots \\ y^{(m-1)}(x_0) &= y_0^{(m-1)}. \end{aligned}$$

Esta ecuación diferencial se puede convertir en un sistema de ecuaciones diferenciales de primer orden, mediante el siguiente cambio de variables:

$$\begin{aligned} u_1 &= y \\ u_2 &= y' \\ u_3 &= y'' \\ &\vdots \\ u_m &= y^{(m-1)} \end{aligned}$$

Entonces la ecuación diferencial se convierte en el siguiente sistema:

$$\begin{aligned} u_1' &= u_2 \\ u_2' &= u_3 \\ u_3' &= u_4 \\ &\vdots \\ u_{m-1}' &= u_m \\ u_m' &= f(x, u_1, u_2, ..., u_m) \\ u_1(x_0) &= y_0 \\ u_2(x_0) &= y_0' \\ u_3(x_0) &= y_0'' \\ &\vdots \\ u_m(x_0) &= y_0^{(m-1)}. \end{aligned}$$

De forma más compacta,

$$\begin{aligned} u' &= F(x, u), \;\; x_0 \leq x \leq b \\ u(x_0) &= \kappa_0, \end{aligned}$$

donde $\kappa_0 = [y_0 \quad y_0' \quad y_0'' \quad ... \quad y_0^{(m-1)}]^{\mathrm{T}}$. Este sistema se puede resolver por los métodos para sistemas de ecuaciones diferenciales de primer orden.

Ejemplo 7.10. Resolver la ecuación diferencial

$$\begin{aligned} y'' &= \frac{4y - xy'}{x^2}, \quad 1 \leq x \leq 2, \\ y(1) &= 3 \\ y'(1) &= 10, \end{aligned}$$

por el método RK4, con $h = 0.2$.

Sean $u_1 = y$, $u_2 = y'$.

$$\begin{aligned} u_1' &= u_2 \\ u_2' &= \frac{4u_1 - xu_2}{x^2}, \quad 1 \leq x \leq 2, \\ u_1(1) &= 3 \\ u_2(1) &= 10. \end{aligned}$$

La solución exacta es $y = 4x^2 - x^{-2}$. Al aplicar el método RK4 se obtiene:

$$\begin{aligned} K^1 &= (2, \ 0.4) \\ K^2 &= (2.04, \ 0.7900826) \\ K^3 &= (2.0790083, \ 0.7678437) \\ K^4 &= (2.1535687, \ 1.0270306) \\ u^1 &= (5.0652642, \ 10.7571472) \\ &\vdots \end{aligned}$$

x_i	$\tilde{u}_1(x_i)$	$\tilde{u}_2(x_i)$	$y(x_i)$
1.0	3.0	10.0	3.0
1.2	5.0652642	10.757147	5.0655556
1.4	7.3293797	11.928367	7.3297959
1.6	9.8488422	13.287616	9.849375
1.8	12.65069	14.742141	12.651358
2.0	15.749173	16.249097	15.75

$\diamond$

7.12 Ecuaciones diferenciales con condiciones de frontera

Una ecuación diferencial de segundo orden con condiciones de frontera se puede escribir de la forma

$$\begin{aligned} y'' &= f(x, y, y'), \quad a \leq x \leq b, \\ y(a) &= y_a \\ y(b) &= y_b. \end{aligned} \tag{7.32}$$

Esta ecuación diferencial se puede convertir en un sistema de dos ecuaciones diferenciales, pero para obtener su solución numérica se presenta un inconveniente: se debería conocer el valor $y'_a = y'(a)$. Esta dificultad se supera mediante el **método del disparo** (*shooting*).

Como no se conoce y'_a, se le asigna un valor aproximado inicial. Puede ser

$$y'_a \approx \frac{y_b - y_a}{b - a}.$$

Con este valor inicial se busca la solución numérica, hasta obtener

$$\tilde{y}(b) = \tilde{y}(b, y'_a).$$

Este valor debería ser el valor conocido y_b. Si no coinciden, es necesario modificar la suposición de y'_a hasta obtener el resultado deseado. Si $\tilde{y}(b, y'_a) < y_b$, entonces se debe aumentar la pendiente inicial del disparo. De manera análoga, si $\tilde{y}(b, y'_a) > y_b$, se debe disminuir la pendiente inicial del disparo. Lo anterior se puede presentar como la solución de una ecuación:

$$\varphi(y'_a) = y_b - \tilde{y}(b, y'_a) = 0.$$

Esta ecuación se puede resolver, entre otros métodos, por el de la secante o el de bisección.

Para facilitar la presentación del método se considera el problema $P(v)$,

donde:

$v =$ aproximación de y'_a,
$n =$ número de intervalos para la solución numérica,
$\tilde{y} = (\tilde{y}_0, \tilde{y}_1, ..., \tilde{y}_n) =$ solución numérica del siguiente problema:

$$\begin{aligned} y' &= f(x,y), \quad a \le x \le b \\ y(a) &= y_a \\ y'(a) &= v, \end{aligned} \qquad \mathrm{P}(v)$$

$$\varphi(v) = y_b - \tilde{y}_n = y_b - \tilde{y}(b,v). \tag{7.33}$$

Se desea encontrar v^* tal que $\varphi(v^*) = 0$. Entonces la solución numérica de $P(v^*)$ es la solución numérica de (7.32). Si se aplica el método de la secante para resolver la ecuación $\varphi(v) = 0$, el algoritmo es el siguiente:

MÉTODO DEL DISPARO

datos: f, a, b, y_a, y_b, ε, `maxit`, ε_0
$\varepsilon_r = \max\{1, |y_b|\}\ \varepsilon$
$v_0 = (y_b - y_a)/(b-a)$
$\tilde{y} =$ solución numérica de $\mathrm{P}(v_0)$
$\varphi_0 = y_b - \tilde{y}_n$
si $|\varphi_0| \le \varepsilon_r$ **ent parar**
$v_1 = v_0 + \varphi_0/(b-a)$
$\tilde{y} =$ solución numérica de $\mathrm{P}(v_1)$
$\varphi_1 = y_b - \tilde{y}_n$
si $|\varphi_1| \le \varepsilon_r$ **ent parar**
para $k = 1, ...,$`maxit`
    $\delta = \varphi_1 - \varphi_0$
    si $|\delta| \le \varepsilon_0$ **ent parar**
    $v_2 = v_1 - \varphi_1(v_1 - v_0)/\delta$
    $\tilde{y} =$ solución numérica de $\mathrm{P}(v_2)$
    $\varphi_2 = y_b - \tilde{y}_n$
    si $|\varphi_2| \le \varepsilon_r$ **ent parar**
    $v_0 = v_1$, $v_1 = v_2$, $\varphi_0 = \varphi_1$, $\varphi_1 = \varphi_2$
fin-para
OJO: no hubo convergencia.

Ejemplo 7.11. Resolver la ecuación diferencial

$$y'' = \frac{2\cos(2x) - y' - 4x^2 y}{x^2}, \quad 0.2 \le x \le 0.7$$
$$y(0.2) = 0.3894183$$
$$y(0.7) = 0.9854497,$$

con $h = 0.1$ y utilizando RK4 para la solución del sistema de ecuaciones diferenciales asociado.

La primera aproximación de $y'(a)$ es

$$v_0 = (0.9854497 - 0.3894183)/(0.7 - 0.2) = 1.19206278$$

Al resolver numéricamente el problema P(1.19206278) se obtiene:

$$\tilde{y}_5 = 0.94935663.$$

El disparo resultó muy bajo.

$$\varphi_0 = 0.03609310$$
$$v_1 = 1.19206278 + 0.03609310/(0.7 - 0.5) = 1.26424897$$

Al resolver numéricamente el problema P(1.26424897) se obtiene:

$$\tilde{y}_5 = 0.95337713$$
$$\varphi_1 = 0.03207260$$

Primera iteración del método de la secante:

$$v_2 = 1.84009748$$

Al resolver numéricamente el problema P(1.84009748) se obtiene:

$$\tilde{y}_5 = 0.98544973$$

Este disparo fue preciso (no siempre se obtiene la solución con una sola iteración de la secante). El último vector $\tilde{y}$ es la solución. La solución exacta es $y = \text{sen}(2x)$.

x_i	$\tilde{y}(x_i)$	$y(x_i)$
0.2	0.3894183	0.3894183
0.3	0.5647741	0.5646425
0.4	0.7174439	0.7173561
0.5	0.8415217	0.8414710
0.6	0.9320614	0.9320391
0.7	0.9854497	0.9854497

$\diamond$

7.13 Ecuaciones diferenciales lineales con condiciones de frontera

Una ecuación diferencial lineal de segundo orden con condiciones de frontera se puede escribir de la forma

$$\begin{aligned} p(x)y'' + q(x)y' + r(x)y = s(x), \quad a \le x \le b, \\ y(a) = y_a \\ y(b) = y_b. \end{aligned} \tag{7.34}$$

Obviamente esta ecuación se puede resolver por el método del disparo, pero, dada la linealidad, se puede resolver usando aproximaciones numéricas (diferencias finitas) para y' y y''.

El intervalo $[a, b]$ se divide en $n \ge 2$ subintervalos de tamaño $h = (b-a)/n$. Los puntos x_i están igualmente espaciados ($x_i = a + ih$). Se utilizan las siguientes aproximaciones y la siguiente notación:

$$\begin{aligned} y_i'' &\approx \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} \\ y_i' &\approx \frac{-y_{i-1} + y_{i+1}}{2h} \\ p_i &:= p(x_i) \\ q_i &:= q(x_i) \\ r_i &:= r(x_i) \\ s_i &:= s(x_i). \end{aligned}$$

Entonces:

$$p_i \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + q_i \frac{-y_{i-1} + y_{i+1}}{2h} + r_i y_i = s_i, \quad i = 1, ..., n-1.$$

Es decir, se tiene un sistema de $n-1$ ecuaciones con $n-1$ incógnitas, y_1, y_2, ..., y_{n-1}.

$$2p_i\frac{y_{i-1}-2y_i+y_{i+1}}{2h^2}+hq_i\frac{-y_{i-1}+y_{i+1}}{2h^2}+\frac{2h^2r_iy_i}{2h^2}=\frac{2h^2s_i}{2h^2}$$
$$(2p_i-hq_i)y_{i-1}+(-4p_i+2h^2r_i)y_i+(2p_i+hq_i)y_{i+1}=2h^2s_i$$

Este sistema es tridiagonal.

$$\begin{bmatrix} d_1 & u_1 & & & & & \\ l_1 & d_2 & u_2 & & & & \\ & l_2 & d_3 & u_3 & & & \\ & & & & & & \\ & & & & l_{n-3} & d_{n-2} & u_{n-2} \\ & & & & & l_{n-2} & d_{n-1} \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \\ y_{n-2} \\ y_{n-1} \end{bmatrix}=\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \\ \beta_{n-2} \\ \beta_{n-1} \end{bmatrix}, \qquad (7.35)$$

donde

$$\begin{aligned} d_i &= -4p_i+2h^2r_i, && i=1,...,n-1,\\ u_i &= 2p_i+hq_i, && i=1,...,n-2,\\ l_i &= 2p_{i+1}-hq_{i+1}, && i=1,...,n-2,\\ \beta_1 &= 2h^2s_1-(2p_1-hq_1)y_a,\\ \beta_i &= 2h^2s_i, && i=2,...,n-2,\\ \beta_{n-1} &= 2h^2s_{n-1}-(2p_{n-1}+hq_{n-1})y_b. \end{aligned}$$

Ejemplo 7.12. Resolver por diferencias finitas la ecuación diferencial

$$\begin{aligned} x^2y''+y'+4x^2y &= 2\cos(2x), \quad 0.2\le x\le 0.7\\ y(0.2) &= 0.3894183\\ y(0.7) &= 0.9854497, \end{aligned}$$

con $n=5$, es decir, $h=0.1$.

Al calcular los coeficientes del sistema tridiagonal se obtiene:

$$
\begin{aligned}
d_1 &= -4p_1 + 2h^2 r_1 \\
d_1 &= -4(0.3)^2 + 2(0.1)^2 4(0.3)^2 = -0.3528 \\
u_1 &= 2p_1 + hq_1 \\
u_1 &= 2(0.3)^2 + 0.1(1) = 0.28 \\
l_1 &= 2p_2 - hq_2 \\
l_1 &= 2(0.4)^2 - 0.1(1) = 0.22 \\
d &= (-0.3528,\ -0.6272,\ -0.98,\ -1.4112), \\
u &= (0.28,\ 0.42,\ 0.6), \\
l &= (0.22,\ 0.4,\ 0.62), \\
\beta &= (0.00186,\ 0.0278683,\ 0.0216121,\ -0.7935745).
\end{aligned}
$$

Su solución es

$$(y_1, y_2, y_3, y_4) = (0.5628333,\ 0.7158127,\ 0.8404825,\ 0.9315998).$$

x_i	$\tilde{y}(x_i)$	$y(x_i)$
0.2	0.3894183	0.3894183
0.3	0.5628333	0.5646425
0.4	0.7158127	0.7173561
0.5	0.8404825	0.8414710
0.6	0.9315998	0.9320391
0.7	0.9854497	0.9854497

$\diamond$

Ejercicios

Escoja varias ecuaciones diferenciales (o sistemas de ecuaciones diferenciales) de las que conozca la solución exacta. Fije el intervalo de trabajo. Determine qué métodos puede utilizar. Aplique varios de ellos. Compare los resultados. Cambie el tamaño del paso. Compare de nuevo.

Un procedimiento adecuado para obtener las ecuaciones diferenciales consiste en partir de la solución (una función cualquiera) y construir la ecuación diferencial.

La aplicación de los métodos se puede hacer de varias maneras: a mano con ayuda de una calculadora; parte a mano y parte con ayuda de software para matemáticas como Scilab o Matlab; haciendo un programa, no necesariamente muy sofisticado, para cada método.

A continuación se presentan algunos ejemplos sencillos.

7.1

$$\begin{aligned} y' &= e^x - \frac{y}{x} \\ y(1) &= 0. \end{aligned}$$

Su solución es $y = e^x - \dfrac{e^x}{x}$.

7.2

$$\begin{aligned} y_1' &= 2y_1 + y_2 + 3 \\ y_2' &= 4y_1 - y_2 + 9 \\ y_1(0) &= -3 \\ y_2(0) &= 5. \end{aligned}$$

Su solución es $y_1(t) = -e^{-2t} - 2,\ \ y_2(t) = 4e^{-2t} + 1.$

7.3

$$\begin{aligned} y'' &= \frac{2}{x(2-x)}y' \\ y(1) &= -2 \\ y'(1) &= 1. \end{aligned}$$

Su solución es $y = -2\ln(2-x) - x - 1$. Tenga especial cuidado con el intervalo de trabajo.

7.4

$$\begin{aligned} y''' + y'' + y' + y &= 4e^x \\ y(0) &= 1 \\ y'(0) &= 2 \\ y''(0) &= 1 \\ y'''(0) &= 0. \end{aligned}$$

Su solución es $y = e^x + \operatorname{sen}(x)$.

7.5

$$\begin{aligned} y''y &= e^{2x} - \operatorname{sen}^2(x) \\ y(0) &= 1 \\ y(\pi) &= e^{\pi}. \end{aligned}$$

Su solución es $y = e^x + \operatorname{sen}(x)$.

7.6

$$\begin{aligned} y'' + e^{-x}y' + y &= 2e^x + 1 + e^{-x}\cos(x) \\ y(0) &= 1 \\ y(\pi) &= e^{\pi}. \end{aligned}$$

Su solución es $y = e^x + \operatorname{sen}(x)$.

8

Ecuaciones diferenciales parciales

8.1 Generalidades

Sea $u = u(x, y)$ una función de dos variables con derivadas parciales de orden dos. Una ecuación diferencial se llama cuasi-lineal si es de la forma

$$Au_{xx} + Bu_{xy} + Cu_{yy} = \varphi(x, y, u, u_x, u_y),$$

dosnde A, B y C son constantes. Hay tres tipos de ecuaciones cuasi-lineales.

$$\begin{array}{ll} \textbf{elíptica} & \text{si } B^2 - 4AC < 0, \\ \textbf{parabólica} & \text{si } B^2 - 4AC = 0, \\ \textbf{hiperbólica} & \text{si } B^2 - 4AC > 0. \end{array}$$

Un ejemplo típico de una ecuación elíptica es la ecuación de Poisson

$$\nabla^2 u = u_{xx} + u_{yy} = f(x, y).$$

Un caso particular es la ecuación de Laplace

$$u_{xx} + u_{yy} = 0.$$

Un ejemplo típico de una ecuación parabólica es la ecuación unidimensional del calor

$$u_t = c^2 u_{xx}.$$

Un ejemplo típico de una ecuación hiperbólica es la ecuación de onda

$$u_{tt} = c^2 u_{xx}.$$

Hay dos grupos importantes de métodos numéricos para EDP: diferencias finitas y elementos finitos. En el primero las derivadas parciales se aproximan por medio de diferencias finitas. Las tres secciones siguientes tratan sobre métodos de diferencias finitas.

8.2 Elípticas: ecuación de Poisson

Consideraremos un caso particular cuando el dominio es un rectangulo,

$$\begin{aligned} \Omega &= \{(x,y) : a < x < b,\ c < y < d\}, \\ \partial\Omega &= \text{frontera de } \Omega. \end{aligned}$$

La ecuación de Poisson con condiciones de frontera de Dirichlet es la siguiente:

$$\begin{aligned} \triangle u(x,y) &= f(x,y) \text{ en } \Omega, \\ u(x,y) &= g(x,y) \text{ en } \partial\Omega. \end{aligned} \tag{8.1}$$

Hay condiciones de frontera que utilizan derivadas con respecto al vector normal en la frontera. Estas condiciones se llaman condiciones de Neumann.

Resolver numéricamente la ecuación diferencial consiste en obtener aproximaciones de $u(x_i, y_j)$, donde los puntos (x_i, y_j) están en Ω. De manera más precisa, sean

$$\begin{aligned} n_x &\in \mathbb{Z},\ n_x \geq 1, \\ n_y &\in \mathbb{Z},\ n_y \geq 1, \\ h_x &= \frac{b-a}{n_x+1}, \\ h_y &= \frac{d-c}{n_y+1}, \\ x_i &= a + ih_x, \quad i = 1, ..., n_x, \\ y_j &= c + jh_y, \quad j = 1, ..., n_y, \\ u_{ij} &\approx u(x_i, y_j), \quad i = 1, ...n_x, \quad j = 1, ..., n_y. \end{aligned}$$

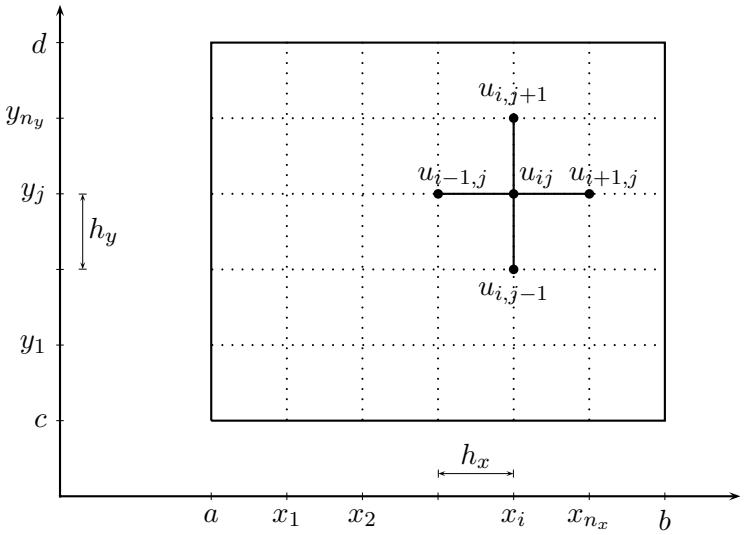


Figura 8.1: División del rectángulo

Usando la aproximación

$$\varphi''(t) \approx \frac{\varphi(t+h) - 2\varphi(t) + \varphi(t-h)}{h^2}$$

se obtiene

$$\triangle u(x_i, y_j) \approx \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h_y^2}. \tag{8.2}$$

Sea $\eta = h_x/h_y$.

$$\triangle u(x_i, y_j) \approx \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2} + \eta^2 \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h_x^2}$$

$$\triangle u(x_i, y_j) \approx \frac{u_{i+1,j} + u_{i-1,j} + \eta^2 u_{i,j+1} + \eta^2 u_{i,j-1} - (2 + 2\eta^2)u_{ij}}{h_x^2}. \tag{8.3}$$

En el caso particular cuando $h = h_x = h_y$

$$\triangle u(x_i, y_j) \approx \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1}u_{i,j-1} - 4u_{ij}}{h^2}. \tag{8.4}$$

Al aplicar la aproximación (8.3) en (8.1), y cambiando el signo aproximación por el signo de igualdad, se obtiene

$$-u_{i+1,j} - u_{i-1,j} - \eta^2 u_{i,j+1} - \eta^2 u_{i,j-1} + (2 + 2\eta^2)u_{ij} = -h_x^2 f_{ij}, \tag{8.5}$$

donde $f_{ij} = f(x_i, y_j)$ son valores conocidos. Al considerar los $n_x n_y$ puntos de la malla se obtiene un sistema de $n_x n_y$ ecuaciones con $n_x n_y$ incógnitas. Para simplificar la notación, sean

$$\begin{aligned} n &= n_x \\ m &= n_y \\ N &= nm \\ h &= h_x \\ \eta &= \frac{h}{h_y} \\ \rho &= \eta^2 \\ \sigma &= 2 + 2\eta^2 \\ \alpha_j &= g(a, y_j) \\ \beta_j &= g(b, y_j) \\ \gamma_i &= g(x_i, c) \\ \delta_i &= g(x_i, d) \end{aligned}$$

Entonces

$$-u_{i+1,j} - u_{i-1,j} - \rho u_{i,j+1} - \rho u_{i,j-1} + \sigma u_{ij} = -h^2 f_{ij} \tag{8.6}$$

Utilizaremos el siguiente orden para los puntos: primero los puntos de la primera fila (la fila horizontal inferior), en seguida los puntos de la segunda fila, ..., y finalmente los puntos de la fila superior. En cada fila el orden es el usual, de izquierda a derecha.

En este orden se plantean la ecuaciones: la ecuación en (x_1, y_1), en (x_2, y_1),

..., en (x_n, y_1), en (x_1, y_2), ... Para las variables utilizaremos el mismo orden

$$\begin{aligned}
\xi_1 &= u_{11} \\
\xi_2 &= u_{21} \\
&\vdots \\
\xi_n &= u_{n1} \\
\xi_{n+1} &= u_{12} \\
\xi_{n+2} &= u_{22} \\
&\vdots \\
\xi_{2n} &= u_{n2} \\
&\vdots \\
\xi_N &= u_{nm}
\end{aligned}$$

Con el anterior orden para las variables la igualdad (8.6) se reescribe así:

$$-\rho u_{i,j-1} - u_{i-1,j} + \sigma u_{ij} - u_{i+1,j} - \rho u_{i,j+1} = -h^2 f_{ij}$$

El sistema de N ecuaciones con N incógnitas se escribe simplemente:

$$A\xi = v. \tag{8.7}$$

En alguno de los siguientes cuatro casos: $i = 1$, $i = n$, $j = 1$ y $j = m$, alguno(s) de los valores u_{kl} corresponde al valor de u en la frontera. En este caso se utilizan las condiciones de frontera, es decir, los valores de g en el punto de frontera específico. Como son valores conocidos, entonces pasan al lado derecho de la igualdad. A continuación están algunas de las igualdades.

Al plantear la ecuación en el punto (x_1, y_1) se obtiene:

$$-\rho u_{10} - u_{01} + \sigma u_{11} - u_{21} - \rho u_{12} = -h^2 f_{11}.$$

Es necesario cambiar u_{10} por el valor conocido γ_1 y cambiar u_{01} por el valor conocido α_1. Utilizando la notación ξ_k se obtiene:

$$\sigma\xi_1 - \xi_2 - \rho\xi_{n+1} = -h^2 f_{11} + \rho\gamma_1 + \alpha_1.$$

En el punto (x_2, y_1) se obtiene:

$$\begin{aligned}
-\rho u_{20} - u_{11} + \sigma u_{21} - u_{31} - \rho u_{22} &= h^2 - f_{21} \\
-\xi_1 + \sigma\xi_2 - \xi_3 - \rho\xi_{n+2} &= -h^2 f_{21} + \rho\gamma_2.
\end{aligned}$$

En el punto (x_3, y_1) se obtiene:

$$-\rho u_{30} - u_{21} + \sigma u_{31} - u_{41} - \rho u_{32} = -h^2 f_{31}$$
$$-\xi_2 + \sigma\xi_3 - \xi_4 - \rho\xi_{n+3} = -h^2 f_{31} + \rho\gamma_3.$$

En el punto (x_n, y_1) se obtiene:

$$-\rho u_{n0} - u_{n-1,1} + \sigma u_{n1} - u_{n+1,1} - \rho u_{n2} = -h^2 f_{n1}$$
$$-\xi_{n-1} + \sigma\xi_n - \rho\xi_{2n} = -h^2 f_{n1} + \rho\gamma_n + \beta_1.$$

En el punto (x_1, y_2) se obtiene:

$$-\rho u_{11} - u_{02} + \sigma u_{12} - u_{22} - \rho u_{13} = -h^2 f_{12}$$
$$-\rho\xi_1 + \sigma\xi_{n+1} - \xi_{n+2} - \rho\xi_{2n+1} = -h^2 f_{12} + \alpha_2.$$

En el punto (x_3, y_2) se obtiene:

$$-\rho u_{31} - u_{22} + \sigma u_{32} - u_{42} - \rho u_{33} = -h^2 f_{32}$$
$$-\rho\xi_3 - \xi_{n+2} + \sigma\xi_{n+3} - \xi_{n+4} - \rho\xi_{2n+3} = -h^2 f_{32}.$$

Si $n = n_x = 3$ y $m = n_y = 4$, la matriz A tiene la siguiente forma:

$$A = \begin{bmatrix}
\sigma & -1 & 0 & -\rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & \sigma & -1 & 0 & -\rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & \sigma & 0 & 0 & -\rho & 0 & 0 & 0 & 0 & 0 & 0 \\
-\rho & 0 & 0 & \sigma & -1 & 0 & -\rho & 0 & 0 & 0 & 0 & 0 \\
0 & -\rho & 0 & -1 & \sigma & -1 & 0 & -\rho & 0 & 0 & 0 & 0 \\
0 & 0 & -\rho & 0 & -1 & \sigma & 0 & 0 & -\rho & 0 & 0 & 0 \\
0 & 0 & 0 & -\rho & 0 & 0 & \sigma & -1 & 0 & -\rho & 0 & 0 \\
0 & 0 & 0 & 0 & -\rho & 0 & -1 & \sigma & -1 & 0 & -\rho & 0 \\
0 & 0 & 0 & 0 & 0 & -\rho & 0 & -1 & \sigma & 0 & 0 & -\rho \\
0 & 0 & 0 & 0 & 0 & 0 & -\rho & 0 & 0 & \sigma & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\rho & 0 & -1 & \sigma & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\rho & 0 & -1 & \sigma
\end{bmatrix}$$

Se puede observar que A es una matriz simétrica, tridiagonal por bloques, de tamaño $m \times m$ bloques, donde cada bloque es de tamaño $n \times n$.

$$A = \begin{bmatrix}
D & -\rho I_n & 0 & & \\
-\rho I_n & D & -\rho I_n & & \\
0 & -\rho I_n & D & & \\
 & & & D & -\rho I_n \\
 & & & -\rho I_n & D
\end{bmatrix}.$$

D es una matriz simétrica tridiagonal de tamaño $n \times n$.

$$D = \begin{bmatrix} \sigma & -1 & 0 & & \\ -1 & \sigma & -1 & & \\ 0 & -1 & \sigma & & \\ & & & \sigma & -1 \\ & & & -1 & \sigma \end{bmatrix}.$$

A es de diagonal positiva dominante. En la mayoría de las filas

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| = 2 + 2\rho = a_{ii}.$$

En algunas filas

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| < a_{ii}.$$

Para resolver $A\xi = v$ se puede utilizar el método de Gauss si m y n son pequeños. Si N es muy grande no se puede almacenar completamente A como matriz densa y, además, el tiempo de cálculo se vuelve muy grande. Hay varios métodos que pueden ser más eficientes para N grande, algunos son específicos para la ecuación de Poisson en un rectángulo. Por ejemplo se puede utilizar el método de Gauss Seidel o el de sobrerrelajación. Estos dos métodos se pueden implementar sin almacenar los elementos no nulos de A. Conociendo m, n, σ y ρ se tiene toda la información sobre A. También se pueden utilizar métodos basados en la FFT (Fast Fourier Transform). Otros métodos son: el de reducción cíclica, el método FACR (Fourier Analysis Cyclic Reduction) y el método de doble barrido de Cholesky.

Ejemplo 8.1. Resolver la ecuación diferencial

$$\begin{aligned} \triangle u &= 6x + 12y, \quad 1 < x < 13, \quad 2 < y < 7 \\ u(a, y) &= 1 + 2y^3 \\ u(b, y) &= 2197 + 2y^3 \\ u(x, c) &= 16 + x^3 \\ u(x, d) &= 686 + x^3 \end{aligned}$$

con $n_x = 3$ y $n_y = 4$.

Entonces $h_x = 3$, $h_y = 1$, $\rho = 9$, $\sigma = 20$,

$v = [235\ \ 2529\ \ 10531\ \ -519\ \ -810\ \ 1353\ \ -505\ \ -918\ \ 1367\ \ 6319\ \ 8235\ \ 16615]^{\mathrm{T}}$.

Al resolver el sistema 12×12 se obtiene

$$u = [118\ \ 397\ \ 1054\ \ 192\ \ 471\ \ 1128\ \ 314\ \ 593\ \ 1250\ \ 496\ \ 775\ \ 1432]^{\mathrm{T}}.$$

La ecuación diferencial es muy sencilla, su solución es $u(x, y) = x^3 + 2y^3$. En este caso, la solución numérica obtenida es exacta. $\diamond$

8.3 Parabólicas: ecuación del calor

La ecuación unidimensional del calor es:

$$\frac{\partial u}{\partial t}(x,t) = c^2 \frac{\partial^2 u}{\partial x^2}(x,t), \quad 0 < x < L, \quad 0 < t\,, \tag{8.8}$$

con las condiciones

$$u(0,t) = v(t), \quad t \geq 0 \tag{8.9}$$
$$u(L,t) = w(t), \quad t \geq 0 \tag{8.10}$$
$$u(x,0) = f(x), \quad 0 \leq x \leq L. \tag{8.11}$$

La función $u(x,t)$ indica la temperatura de una barra uniforme, en la posición x y en el tiempo t. Generalmente las funciones $v(t)$ y $w(t)$ son constantes, es decir, se supone que la temperatura en los extremos de la barra es constante para todo tiempo t.

De manera análoga a las ecuaciones elípticas, se coloca en la región

$$\Omega =]0, L[\times]0, +\infty[$$

una malla determinada por los valores

$$x_i = i\, h_x, \quad i = 0, 1, 2, ..., m$$
$$t_j = j\, h_t, \quad j = 0, 1, 2, ...$$

donde

$$h_x = \frac{L}{m}\,.$$

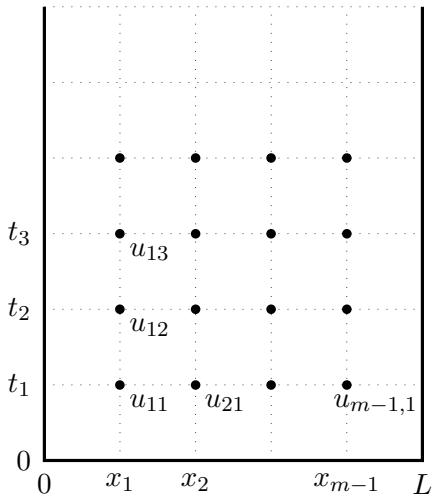


Figura 8.2: Malla para la ecuación del calor

El objetivo es encontrar valores u_{ij}, aproximaciones de los valores $u(x_i, t_j)$. Como se conoce la función u en la frontera de Ω, entonces se conocen los valores:

$$\begin{aligned} &u_{00}, u_{10}, ..., u_{m0}, && t = 0, \\ &u_{01}, u_{02}, ..., u_{0j}, ... && x = 0, \\ &u_{m1}, u_{m2}, ..., u_{mj}, ... && x = L. \end{aligned}$$

Los valores buscados son:

$$\begin{aligned} &u_{11}, u_{21}, ..., u_{m-1,1}, && t = t_1 \\ &u_{12}, u_{22}, ..., u_{m-1,2}, && t = t_2 \\ &u_{13}, u_{23}, ..., u_{m-1,3}, && t = t_3 \\ &\vdots \end{aligned}$$

Cada uno de los paquetes anteriores tiene $m - 1$ valores correspondientes a un tiempo fijo t_j.

Aunque el problema está plantedao para $0 < t < +\infty$, obviamente no se

puede ir hasta infinito. Entonces se toma un valor T adecuado y

$$0 \le t \le T \tag{8.12}$$

$$h_t = \frac{T}{n} \tag{8.13}$$

$$t_j = j\, h_t, \quad j = 0, 1, 2, ..., n. \tag{8.14}$$

8.3.1 Método explícito

La segunda derivada $\partial^2 u/\partial x^2$ se aproxima como en el caso elíptico, la derivada $\partial u/\partial t$ se aproxima hacia adelante:

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_j) \approx \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2} \tag{8.15}$$

$$\frac{\partial u}{\partial t}(x_i, t_j) \approx \frac{u_{i,j+1} - u_{ij}}{h_t} \tag{8.16}$$

Remplazando en (8.8) se obtiene

$$\frac{u_{i,j+1} - u_{ij}}{h_t} = c^2 \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2}$$

$$u_{i,j+1} = \frac{c^2 h_t}{h_x^2}\, u_{i-1,j} + \left(1 - \frac{2c^2 h_t}{h_x^2}\right) u_{ij} + \frac{c^2 h_t}{h_x^2}\, u_{i+1,j}$$

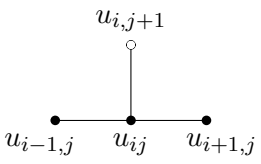
$$u_{i,j+1} = \alpha\, u_{i-1,j} + \beta\, u_{ij} + \alpha\, u_{i+1,j} \tag{8.17}$$

$$\alpha = \frac{c^2 h_t}{h_x^2} \tag{8.18}$$

$$\beta = 1 - 2\alpha. \tag{8.19}$$

En la fórmula (8.15) el error es del orden de $O(h_x^2)$, en (8.16) el error es del orden de $O(h_t)$. El error en (8.17) es del orden de $(h_t + h_x^2)$.

Los valores usados en (8.17) forman la "molécula":



Para calcular los valores u_{11}, u_{21}, ..., $u_{m-1,1}$ se necesitan valores u_{k0}, pero estos son conocidos por corresponder a la condición (8.11). Entonces los valores u_{11}, u_{21}, ..., $u_{m-1,1}$, se calculan sin ningún problema.

Para calcular los valores u_{12}, u_{22}, ..., $u_{m-1,2}$ se necesitan los valores u_{01}, u_{11}, u_{21}, ..., $u_{m-1,1}$, u_{m1}. El primero y el último están dados por las condiciones (8.9) y (8.10); los otros se acaban de calcular. Después, de manera semejante, se calculan los valores u_{i3} y así sucesivamente.

Ejemplo 8.2. Aplicar las fórmulas anteriores a la ecuación diferencial

$$\frac{\partial u}{\partial t}(x,t) = \frac{2}{9}\,\frac{\partial^2 u}{\partial x^2}(x,t), \quad 0 < x < \frac{\pi}{3}, \quad 0 < t \leq 2$$

con las condiciones

$$\begin{aligned} u(0,t) &= 5, \quad t \geq 0 \\ u(\frac{\pi}{3},t) &= 5, \quad t \geq 0 \\ u(x,0) &= \operatorname{sen}(3x) + 5, \quad 0 \leq x \leq \frac{\pi}{3}. \end{aligned}$$

con $m = 10$ y $n = 50$.

La solución exacta de la ecuación diferencial es

$$u(x,t) = e^{-2t}\operatorname{sen}(3x) + 5\,.$$

Al aplicar las fórmulas se obtiene:

$$\begin{aligned} h_x &= \pi/30 = 0.1047198 \\ h_t &= 0.04 \\ \alpha &= 0.8105695 \\ \beta &= -0.6211389 \end{aligned}$$

Para empezar, los valores u_{00}, u_{10}, u_{20}, ..., $u_{10,0}$ son: 5, 5.309017, 5.5877853, 5.809017, 5.9510565, 6, 5.9510565, 5.809017, 5.5877853, 5.309017, 5.

Para $t = t_1 = 0.04$ son datos: $u_{01} = 5$, $u_{10,0} = 5$.

$$\begin{aligned} u_{11} &= \alpha u_{00} + \beta u_{10} + \alpha u_{20} \\ u_{11} &= 5.2844983 \\ u_{21} &= \alpha u_{10} + \beta u_{20} + \alpha u_{30} \\ u_{21} &= 5.5411479 \\ u_{91} &= \alpha u_{80} + \beta u_{90} + \alpha u_{10,0} \\ u_{91} &= 5.2844983 \end{aligned}$$

Para $t = t_2 = 0.08$

$$u_{12} = \alpha u_{01} + \beta u_{11} + \alpha u_{21}$$
$$u_{12} = 5.261925$$

En la tabla siguiente aparecen los valores exactos $u(x_i, 2)$ y los valores obtenidos $u_{i,50}$.

```
5.0000000    5.0000000
5.0056598    9.2699374
5.0107657  - 3.1030622
5.0148177    16.178842
5.0174192  - 8.1110307
5.0183156    18.817809
5.0174192  - 8.1110307
5.0148177    16.178842
5.0107657  - 3.1030622
5.0056598    9.2699374
5.0000000    5.0000000
```

Se observa que los resultados obtenidos por las fórmulas no son buenos. Pero si se utiliza $n = 100$ sí lo son. En la tabla siguiente están los valores teóricos, los obtenidos con $n = 100$ y los errores:

```
5.000000    5.000000    0.000000
5.005660    5.005394    0.000266
5.010766    5.010261    0.000505
5.014818    5.014123    0.000695
5.017419    5.016602    0.000817
5.018316    5.017456    0.000859
5.017419    5.016602    0.000817
5.014818    5.014123    0.000695
5.010766    5.010261    0.000505
5.005660    5.005394    0.000266
5.000000    5.000000    0.000000
```

El ejemplo anterior muestra resultados malos con $n = 50$ y bastante buenos con $n = 100$. Esto tiene una razón: el método con las fórmulas (8.17) es a veces estable, a veces inestable (los errores de redondeo o truncamiento se

propagan exponencialmente). Este método es *condicionalmente estable* (ver [BuF85]). Si

$$\frac{c^2\, h_t}{h_x^2} \leq \frac{1}{2}\,. \tag{8.20}$$

el método es estable.

Fácilmente se comprueba que, en el ejemplo anterior,

$$\frac{c^2\, h_t}{h_x^2} = 0.8105695 \quad \text{si } n = 50$$
$$\frac{c^2\, h_t}{h_x^2} = 0.4052847 \quad \text{si } n = 100$$

8.3.2 Método implícito

La derivada $\partial u/\partial t$ se aproxima hacia atrás:

$$\frac{\partial u}{\partial t}(x_i, t_j) \approx \frac{u_{i,j} - u_{i,j-1}}{h_t} \tag{8.21}$$
$$\frac{\partial^2 u}{\partial x^2}(x_i, t_j) \approx \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2} \tag{8.22}$$

Remplazando en (8.8) se obtiene

$$\frac{u_{i,j} - u_{i,j-1}}{h_t} = c^2 \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2}$$

Si queremos calcular los valores u_{kj}, para $t = t_j$, y conocemos los valores para $t = t_{j-1}$, entonces agrupamos así:

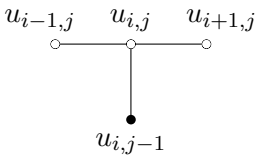
$$-\frac{c^2 h_t}{h_x^2}\, u_{i-1,j} + \left(1 + \frac{c^2 h_t}{h_x^2}\right) u_{ij} - \frac{c^2 h_t}{h_x^2}\, u_{i+1,j} = u_{i,j-1}\,.$$

De manera más compacta:

$$-\alpha\, u_{i-1,j} + \gamma u_{ij} - \alpha u_{i+1,j} = u_{i,j-1} \tag{8.23}$$
$$\alpha = \frac{c^2 h_t}{h_x^2} \tag{8.24}$$
$$\gamma = 1 + 2\alpha. \tag{8.25}$$

La fórmula (8.23), al igual que en el método explícito, tiene un error de orden $O(h_t + h_x^2)$.

Los valores usados en (8.23) forman la "molécula":



Al utilizar (8.23) para los $m-1$ puntos (x_1, t_j), (x_2, t_j), ..., (x_{m-1}, t_j), se obtiene el siguiente sistema de ecuaciones:

$$\begin{bmatrix} \gamma & -\alpha & 0 & 0 & ... & 0 \\ -\alpha & \gamma & -\alpha & 0 & ... & 0 \\ 0 & -\alpha & \gamma & -\alpha & ... & 0 \\ & & & & & \\ & & & & & \\ 0 & 0 & & & -\alpha & \gamma \end{bmatrix} \begin{bmatrix} u_{1j} \\ u_{2j} \\ u_{3j} \\ \\ \\ u_{m-1,j} \end{bmatrix} = \begin{bmatrix} u_{1,j-1} + \alpha u_{0j} \\ u_{2,j-1} \\ u_{3,j-1} \\ \\ \\ u_{m-1,j-1} + \alpha u_{mj} \end{bmatrix} \tag{8.26}$$

Este sistema tridiagonal se puede resolver por cualquier método, pero es más eficiente resolverlo por un método específico para sistemas tridiagonales, ver sección (2.13). Además, como la matriz del sistema es la misma para todas las iteraciones, entonces la factorización LU tridiagonal es la misma para todas las iteraciones y se calcula únicamente una vez. Así, en cada iteración se resuelve el sistema conociendo ya la factorización LU.

Los valores u_{0j} y u_{mj} están dados por los valores $v(t_j)$ y $w(t_j)$ provenientes de las condiciones de frontera.

Ejemplo 8.3. Aplicar este método a la misma ecuación diferencial

$$\frac{\partial u}{\partial t}(x,t) = \frac{2}{9}\frac{\partial^2 u}{\partial x^2}(x,t), \quad 0 < x < \frac{\pi}{3}, \quad 0 < t \leq 2$$

con las condiciones

$$\begin{aligned} u(0,t) &= 5, \quad t \geq 0 \\ u(\frac{\pi}{3},t) &= 5, \quad t \geq 0 \\ u(x,0) &= \text{sen}(3x) + 5, \quad 0 \leq x \leq \frac{\pi}{3}. \end{aligned}$$

con $m = 10$ y $n = 50$.

Al aplicar las fórmulas se obtiene:

$$h_x = \pi/30 = 0.1047198$$
$$h_t = 0.04$$
$$\alpha = 0.8105695$$
$$\gamma = 2.6211389$$

Para empezar, los valores u_{00}, u_{10}, u_{20}, ..., $u_{10,0}$ son: 5, 5.309017, 5.5877853, 5.809017, 5.9510565, 6, 5.9510565, 5.809017, 5.5877853, 5.309017, 5.

Los valores α y γ definen la matriz del sistema (8.26) para todas las iteraciones. Para $t = t_1 = 0.04$, los términos independientes son: 9.3618643, 5.5877853, 5.809017, 5.9510565, 6, 5.9510565, 5.809017, 5.5877853, 9.3618643 .

La solución del sistema es : 5.2863007, 5.5445763, 5.749545, 5.8811429, 5.9264885, 5.8811429, 5.749545, 5.5445763, 5.2863007 . Estos valores corresponden a u_{11}, u_{21}, ..., u_{91}.

La siguiente tabla muestra, para $t = 2$, los valores teóricos, los valores obtenidos por el método y las diferencias:

```
5.000000    5.000000     0.000000
5.005660    5.006792    -0.001132
5.010766    5.012919    -0.002153
5.014818    5.017781    -0.002963
5.017419    5.020903    -0.003484
5.018316    5.021979    -0.003663
5.017419    5.020903    -0.003484
5.014818    5.017781    -0.002963
5.010766    5.012919    -0.002153
5.005660    5.006792    -0.001132
5.000000    5.000000     0.000000
```

Si se considera $n = 100$, los valores para $t = 2$ son:

```
5.000000    5.000000     0.000000
5.005660    5.006315    -0.000655
5.010766    5.012011    -0.001245
5.014818    5.016532    -0.001714
5.017419    5.019434    -0.002015
```

```
5.018316    5.020434   -0.002119
5.017419    5.019434   -0.002015
5.014818    5.016532   -0.001714
5.010766    5.012011   -0.001245
5.005660    5.006315   -0.000655
5.000000    5.000000    0.000000
```

8.3.3 Método de Crank-Nicolson

De acuerdo con las fórmulas (6.29) y (6.30) el valor

$$\delta = \frac{u_{ij} - u_{i,j-1}}{h_t}$$

se puede considerar como la aproximación de $\dfrac{\partial u}{\partial t}(x_i, t_j)$ o bien como la aproximación de $\dfrac{\partial u}{\partial t}(x_i, t_{j-1})$. Es decir,

$$\frac{\partial u}{\partial t}(x_i, t_{j-1}) = \frac{u_{ij} - u_{i,j-1}}{h_t} - \frac{h_t}{2}\frac{\partial^2 u}{\partial t^2}(x_i, \xi), \quad \xi \in [t_{j-1}, t_j],$$
$$\frac{\partial u}{\partial t}(x_i, t_j) = \frac{u_{ij} - u_{i,j-1}}{h_t} + \frac{h_t}{2}\frac{\partial^2 u}{\partial t^2}(x_i, \zeta), \quad \zeta \in [t_{j-1}, t_j].$$

En ambos casos el error es del orden de $O(h_t)$. El mismo valor δ puede ser interpretado de una tercera manera usando (6.32):

$$\frac{\partial u}{\partial t}(x_i, t_{j-1} + h_t/2) = \frac{u_{ij} - u_{i,j-1}}{h_t} - \frac{h_t^2}{24}\frac{\partial^3 u}{\partial t^3}(x_i, \tau), \quad \tau \in [t_{j-1}, t_j],$$

El valor $t_{j-1} + h_t/2$, que será denotado por $t_{j-1/2}$, es el punto medio entre t_{j-1} y t_j. Al plantear la ecuación diferencial en el punto $(x_i, t_{j-1/2})$ tenemos:

$$\frac{\partial u}{\partial t}(x_i, t_{j-1/2}) = c^2 \frac{\partial^2 u}{\partial x^2}(x_i, t_{j-1/2})$$

Ahora remplazaremos $\dfrac{\partial u}{\partial t}$ por δ y $\dfrac{\partial^2 u}{\partial x^2}(x_i, t_{j-1/2})$ por el promedio de aproximaciones de $\dfrac{\partial^2 u}{\partial x^2}$ en dos puntos vecinos:

$$\frac{u_{ij} - u_{i,j-1}}{h_t} = \frac{c^2}{2}\left(\frac{\partial^2 u}{\partial x^2}(x_i, t_{j-1}) + \frac{\partial^2 u}{\partial x^2}(x_i, t_j)\right)$$
$$\frac{u_{ij} - u_{i,j-1}}{h_t} = \frac{c^2}{2}\left(\frac{u_{i-1,j-1} - 2u_{i,j-1} + u_{i+1,j-1}}{h_x^2} + \frac{u_{i-1,j} - 2u_{ij} + u_{i+1,j}}{h_x^2}\right)$$

Esta fórmula tiene un error de orden $O(h_t^2+h_x^2)$. Ahora agruparemos dejando a izquierda los valores buscados, $t = t_j$, y a derecha los que se suponen conocidos, $t = t_{j-1}$:

$$-\rho\, u_{i-1,j} + \mu\, u_{ij} - \rho\, u_{i+1,j} = \rho\, u_{i-1,j-1} + \varphi\, u_{i,j-1} + \rho\, u_{i+1,j-1} \tag{8.27}$$

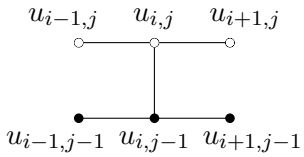
$$\alpha = \frac{c^2 h_t}{h_x^2} \tag{8.28}$$

$$\rho = \frac{\alpha}{2} \tag{8.29}$$

$$\mu = 1 + \alpha \tag{8.30}$$

$$\varphi = 1 - \alpha \tag{8.31}$$

La "molécula" correspondiente a (8.27) es:



Al utilizar (8.27) para $i = 1, 2, ..., m-1$ se obtiene el sistema tridiagonal, que se puede resolver de manera eficiente:

$$\begin{bmatrix} \mu & -\rho & 0 & 0 & ... & 0 \\ -\rho & \mu & -\rho & 0 & ... & 0 \\ 0 & -\rho & \mu & -\rho & ... & 0 \\ & & & & & \\ & & & & & \\ 0 & 0 & & & -\rho & \mu \end{bmatrix} \begin{bmatrix} u_{1j} \\ u_{2j} \\ u_{3j} \\ \\ \\ u_{m-1,j} \end{bmatrix} =$$

$$\begin{bmatrix} \rho u_{0,j-1} + \varphi u_{1,j-1} + \rho u_{2,j-1} + \rho u_{0j} \\ \rho u_{1,j-1} + \varphi u_{2,j-1} + \rho u_{3,j-1} \\ \rho u_{2,j-1} + \varphi u_{3,j-1} + \rho u_{4,j-1} \\ \\ \\ \rho u_{m-2,j-1} + \varphi u_{m-1,j-1} + \rho u_{m,j-1} + \rho u_{mj} \end{bmatrix} \tag{8.32}$$

Ejemplo 8.4. Resolver la misma ecuación diferencial de los dos ejemplos anteriores por el método de Crank-Nicolson, con $m = 10$, $T = 2$, $n = 50$.

$$
\begin{aligned}
h_x &= 0.1047198 \\
h_t &= 0.04 \\
\alpha &= 0.8105695 \\
\rho &= 0.405284 \\
\mu &= 1.8105695 \\
\varphi &= 0.1894305
\end{aligned}
$$

Para empezar, los valores u_{00}, u_{10}, u_{20}, ..., $u_{10,0}$ son: 5, 5.309017, 5.5877853, 5.809017, 5.9510565, 6, 5.9510565, 5.809017, 5.5877853, 5.309017, 5.

Los valores μ y ρ definen la matriz del sistema (8.32) para todas las iteraciones. Para $t = t_1 = 0.04$, los términos independientes son: 7.3231813, 5.5644666, 5.7769216, 5.9133261, 5.9603279, 5.9133261, 5.7769216, 5.5644666, 7.3231813 .

La solución del sistema es : 5.2854339, 5.5429275, 5.7472756, 5.8784752, 5.9236835, 5.8784752, 5.7472756, 5.5429275, 5.2854339.

Estos valores corresponden a u_{11}, u_{21}, ..., u_{91}.

La siguiente tabla muestra, para $t = 2$, los valores teóricos, los valores obtenidos por el método y las diferencias:

```
5.000000    5.000000    0.000000
5.005660    5.005836   -0.000176
5.010766    5.011101   -0.000336
5.014818    5.015280   -0.000462
5.017419    5.017962   -0.000543
5.018316    5.018887   -0.000571
5.017419    5.017962   -0.000543
5.014818    5.015280   -0.000462
5.010766    5.011101   -0.000336
5.005660    5.005836   -0.000176
5.000000    5.000000    0.000000
```

Si se considera $n = 100$, los valores para $t = 2$ son:

```
5.000000    5.000000    0.000000
5.005660    5.005845   -0.000186
```

```
5.010766    5.011119   -0.000353
5.014818    5.015304   -0.000486
5.017419    5.017990   -0.000571
5.018316    5.018916   -0.000601
5.017419    5.017990   -0.000571
5.014818    5.015304   -0.000486
5.010766    5.011119   -0.000353
5.005660    5.005845   -0.000186
5.000000    5.000000    0.000000
```

Los resultados obtenidos por el método de Crank-Nicolson con $n = 50$ son mejores que los obtenidos con el método implícito. Los resultados obtenidos con el método de Crank-Nicolson con $n = 100$ ($h_t^2 = 0.0004$, $h_x^2 = 0.0109662$, $h_t^2 + h_x^2 = 0.0113662$), no mejoran (empeoran ligeramente) los obtenidos con $n = 50$ ($h_t^2 = 0.0016$, $h_x^2 = 0.0109662$, $h_t^2 + h_x^2 = 0.0125662$). En este caso el orden del error depende fundamentalmente de h_x. Si se utiliza el método de Crank-Nicolson con $m = 20$ y $n = 50$ ($h_t^2 = 0.0016$, $h_x^2 = 0.0027416$, $h_t^2 + h_x^2 = 0.0043416$) los resultados mejoran notablemente:

```
5.000000    5.000000    0.000000
5.002865    5.002883   -0.000018
5.005660    5.005694   -0.000035
5.008315    5.008366   -0.000051
5.010766    5.010831   -0.000066
5.012951    5.013030   -0.000079
5.014818    5.014908   -0.000091
5.016319    5.016419   -0.000100
5.017419    5.017526   -0.000107
5.018090    5.018201   -0.000111
5.018316    5.018428   -0.000112
5.018090    5.018201   -0.000111
5.017419    5.017526   -0.000107
5.016319    5.016419   -0.000100
5.014818    5.014908   -0.000091
5.012951    5.013030   -0.000079
5.010766    5.010831   -0.000066
5.008315    5.008366   -0.000051
5.005660    5.005694   -0.000035
5.002865    5.002883   -0.000018
5.000000    5.000000    0.000000
```

8.4 Hiperbólicas: ecuación de onda

Consideramos la siguiente ecuación

$$\frac{\partial^2 u}{\partial t^2}(x,t) = c^2 \frac{\partial^2 u}{\partial x^2}(x,t), \quad 0 < x < L, \quad 0 < t\,, \tag{8.33}$$

con las condiciones

$$u(0,t) = a, \quad t \geq 0, \tag{8.34}$$
$$u(L,t) = b, \quad t \geq 0, \tag{8.35}$$
$$u(x,0) = f(x), \quad 0 \leq x \leq L, \tag{8.36}$$
$$\frac{\partial u}{\partial t}(x,0) = g(x), \quad 0 \leq x \leq L. \tag{8.37}$$

Esta ecuación describe el movimiento en el tiempo de una cuerda vibrante, de longitud L, fija en los extremos y de la que se conoce la posición inicial y la velocidad inicial. Generalmente los valores constantes a y b son iguales y nulos.

8.4.1 Método explícito

La región es la misma de la ecuación del calor y se divide exactamente de la misma forma. Sea T un tiempo adecuado:

$$\begin{aligned} h_x &= \frac{L}{m} \\ h_t &= \frac{T}{n} \\ x_i &= i\,h_x, \quad i = 0,1,2,...,m \\ t_j &= j\,h_t, \quad j = 0,1,2,...,n. \end{aligned}$$

Se desea conocer los valores u_{ij}, buenas aproximaciones de $u(x_i,t_j)$. Se utiliza la ecuación diferencial en el punto (x_i,t_j):

$$\frac{\partial^2 u}{\partial t^2}(x_i,t_j) = c^2 \frac{\partial^2 u}{\partial x^2}(x_i,t_j) \tag{8.38}$$

Aproximando las segundas derivadas por diferencias finitas en el punto (x_i, t_j) se obtiene:

$$\frac{u_{i,j-1} - 2u_{ij} + u_{i,j+1}}{h_t^2} = c^2 \, \frac{u_{i-1,j} - 2u_{ij} + u_{i+1,j}}{h_x^2} \, .$$

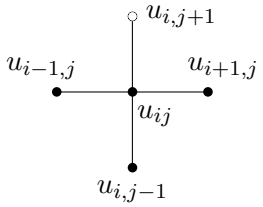
Si se suponen conocidos los valores para $t = t_j$ y para $t = t_{j-1}$, entonces se puede obtener $u_{i,j+1}$:

$$u_{i,j+1} = \beta u_{ij} + \alpha(u_{i-1,j} + u_{i+1,j}) - u_{i,j-1} \tag{8.39}$$

$$\alpha = \frac{c^2 \, h_t^2}{h_x^2} \tag{8.40}$$

$$\beta = 2 - 2\alpha \tag{8.41}$$

La molécula es:



La fórmula (8.39), con error de orden $O(h_x^2 + h_t^2)$, se puede aplicar fácilmente, salvo para la obtención de los valores u_{i1} ya que sería necesario conocer los valores $u_{i,-1}$. Aproximaciones de estos valores se obtienen utilizando las condiciones (8.37) sobre la velocidad inicial, mediante la siguiente aproximación cuyo error es del orden de $O(h_t^2)$,

$$g_i = g(x_i) \approx \frac{u_{i1} - u_{i,-1}}{2h_t} \, ,$$

$$u_{i,-1} = u_{i1} - 2h_t \, g_i \, , \tag{8.42}$$

Remplazando en (8.39) para $j = 0$ y teniendo en cuenta que $u_{k0} = f_k = f(x_k)$,

$$\begin{aligned} u_{i1} &= \beta f_i + \alpha(f_{i-1} + f_{i+1}) - (u_{i1} - 2h_t g_i) \\ 2u_{i1} &= \beta f_i + \alpha(f_{i-1} + f_{i+1}) + 2h_t g_i \\ u_{i1} &= \frac{\beta}{2} f_i + \frac{\alpha}{2}(f_{i-1} + f_{i+1}) + h_t g_i, \quad i = 1, 2, ..., m-1 \end{aligned} \tag{8.43}$$

Una vez calculados los valores u_{i1} por medio de (8.43), se utiliza (8.39) para $j = 1, 2, ..., n$, teniendo en cuenta que $u_{0j} = a$ y $u_{mj} = b$.

Ejemplo 8.5. Resolver la ecuación diferencial

$$\frac{\partial^2 u}{\partial t^2}(x,t) = 3\,\frac{\partial^2 u}{\partial x^2}(x,t), \quad 0 < x < 2, \quad 0 < t\,,$$

con las condiciones

$$\begin{aligned} u(0,t) &= 0, \quad t \geq 0, \\ u(2,t) &= 0, \quad t \geq 0, \\ u(x,0) &= \frac{1}{4}\,x(2-x), \quad 0 \leq x \leq 2, \\ \frac{\partial u}{\partial t}(x,0) &= 0, \quad 0 \leq x \leq 2, \end{aligned}$$

utilizando $T = 12$, $m = 10$, $n = 300$.

Se puede comprobar que la solución exacta de esta ecuación diferencial es

$$\begin{aligned} u(x,t) &= \sum_{k=1}^{\infty} A_k \cos\left(\frac{k\pi c\,t}{2}\right) \operatorname{sen}\left(\frac{k\pi x}{2}\right), \\ A_k &= \frac{4(1-(-1)^k)}{k^3\pi^3}, \\ c &= \sqrt{3}. \end{aligned}$$

Para la solución numérica

$$\begin{aligned} h_x &= 0.2 \\ h_t &= 0.04 \\ \alpha &= 0.12 \\ \beta &= 1.76 \end{aligned}$$

Los valores inciales u_{i0} son: 0, 0.09, 0.16, 0.21, 0.24, 0.25, 0.24, 0.21, 0.16, 0.09, 0.

Para calcular los valores u_{i1} se utiliza (8.43) y se obtiene: 0, 0.0888, 0.1588, 0.2088, 0.2388, 0.2488, 0.2388, 0.2088, 0.1588, 0.0888, 0.

Los demás valores u_{ik} se calculan usando (8.39). Así los valores u_{i2} son: 0, 0.085344, 0.1552, 0.2052, 0.2352, 0.2452, 0.2352, 0.2052, 0.1552, 0.085344, 0.

A continuación aparecen: los valores x_i, los valores $u(x_i, 12)$ exactos y los valores aproximados obtenidos por el método:

```
0.00   0.000000   0.000000
0.20   0.021539   0.032374
0.40   0.043078   0.057421
0.60   0.064617   0.073748
0.80   0.086097   0.080376
1.00   0.096097   0.080269
1.20   0.086097   0.080376
1.40   0.064617   0.073748
1.60   0.043078   0.057421
1.80   0.021539   0.032374
2.00   0.000000   0.000000
```

Este método presenta problemas de inestabilidad. Se puede garantizar la estabilidad (no la exactitud) si

$$c\,\frac{h_t}{h_x} \leq 1. \tag{8.44}$$

En el ejemplo anterior $ch_t/h_x = 0.3464$. Si se hubiera aplicado el método con $n = 100$, $ch_t/h_x = 1.0392$, los resultados serían:

```
0.00   0.000000                    0.
0.20   0.021539  -3366402376293274.
0.40   0.043078   6403277832890416.
0.60   0.064617  -8813355840944206.
0.80   0.086097  10360721173025462.
1.00   0.096097 -10893906929301864.
1.20   0.086097  10360721173025462.
1.40   0.064617  -8813355840944206.
1.60   0.043078   6403277832890416.
1.80   0.021539  -3366402376293274.
2.00   0.000000                    0.
```

8.4.2 Método implícito

Consideremos la ecuación diferencial en el punto (x_i, t_j),

$$\frac{\partial^2 u}{\partial t^2}(x_i, t_j) = c^2 \frac{\partial^2 u}{\partial x^2}(x_i, t_j)\,,$$

pero cambiando la doble derivada parcial con respecto a x, en el punto (x_i, t_j), por el promedio de la derivada en los puntos vecinos (x_i, t_{j-1}) y (x_i, t_{j+1}):

$$\frac{\partial^2 u}{\partial t^2}(x_i, t_j) = \frac{c^2}{2}\left(\frac{\partial^2 u}{\partial x^2}(x_i, t_{j-1}) + \frac{\partial^2 u}{\partial x^2}(x_i, t_{j+1})\right)$$

Ahora utilizamos aproximación por diferencias finitas:

$$\frac{u_{i,j-1} - 2u_{ij} + u_{i,j+1}}{h_t^2} = \frac{c^2}{2}\Big(\frac{u_{i-1,j-1} - 2u_{i,j-1} + u_{i+1,j-1}}{h_x^2} + \frac{u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1}}{h_x^2}\Big)$$

Ahora dejamos a la izquierda los valores desconocidos y a la derecha los que son conocidos:

$$\begin{aligned} 2u_{i,j-1} - 4u_{ij} + 2u_{i,j+1} &= \alpha(u_{i-1,j-1} - 2u_{i,j-1} + u_{i+1,j-1} \\ &\quad + u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1}) \\ -\alpha u_{i-1,j+1} + \gamma u_{i,j+1} - \alpha u_{i+1,j+1} &= 4u_{ij} - \gamma u_{i,j-1} + \alpha(u_{i-1,j-1} + u_{i+1,j-1}) \end{aligned} \tag{8.45}$$

$$\alpha = \frac{c^2 h_t^2}{h_x^2} \tag{8.46}$$

$$\gamma = 2 + 2\alpha \tag{8.47}$$

Aplicando la igualdad (8.45) en los puntos (x_1, t_j), (x_2, t_j), ..., (x_{m-1}, t_j), se obtiene el siguiente sistema tridiagonal, de tamaño $(m-1) \times (m-1)$:

$$\begin{bmatrix} \gamma & -\alpha & & & & \\ -\alpha & \gamma & -\alpha & & & \\ 0 & -\alpha & \gamma & -\alpha & & \\ & & & & & \\ & & & & & \\ 0 & 0 & 0 & & -\alpha & \gamma \end{bmatrix} \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \vdots \\ u_{m-2,j+1} \\ u_{m-1,j+1} \end{bmatrix} =$$

$$\begin{bmatrix} 4u_{1j} - \gamma u_{1,j-1} + \alpha(a + u_{2,j-1}) \\ 4u_{2j} - \gamma u_{2,j-1} + \alpha(u_{1,j-1} + u_{3,j-1}) \\ \vdots \\ 4u_{m-2,j} - \gamma u_{m-2,j-1} + \alpha(u_{m-3,j-1} + u_{m-1,j-1}) \\ 4u_{m-1,j} - \gamma u_{m-1,j-1} + \alpha(u_{m-2,j-1} + b) \end{bmatrix} \tag{8.48}$$

Este sistema tridiagonal se puede resolver eficientemente. Para empezar, también es necesario calcular los valores u_{i1} por medio de (8.43). Después es necesario resolver $n-1$ veces el sistema (8.48). Este método implícito no es inestable.

Ejercicios

8.1 Considere la ecuación diferencial $\dfrac{\partial^2 u}{\partial t^2}(x,t) = \dfrac{\partial^2 u}{\partial x^2}(x,t)$, para $0 < x < 1$ y $0 < t$ con las condiciones $u(0,t) = u(1,t) = 0$, $u(x,0) = \operatorname{sen}(\pi x)$, $\dfrac{\partial u}{\partial t}(x,0) = 0$. La solución exacta es $u(x,t) = \operatorname{sen}(\pi x)\cos(\pi t)$. Obtenga la solución aproximada para $t = 3$ con $m = 10$, $n = 60$ y con $n = 20$. Compare con la solución exacta.

9

Valores propios

9.1 Preliminares

Sea $A \in \mathbb{R}^{n\times n}$, una matriz cuadrada real, un número λ, real o complejo, es un *valor propio* de A si existe un vector columna real o complejo no nulo $v \in \mathbb{C}^{n\times 1}$ tal que

$$Av = \lambda v.$$

En ese caso se dice que v es un *vector propio* asociado al valor propio λ. Fácilmente se comprueba que si $\alpha \neq 0$, entonces también αv es un vector propio asociado a λ. Generalmente se usan más los vectores propios normalizados, es decir, $||v||_2 = 1$.

Ejemplo 9.1. Sea

$$A = \begin{bmatrix} 8 & 2 & 1 \\ 1 & 7 & 3 \\ 1 & 1 & 6 \end{bmatrix}.$$

Como

$$\begin{bmatrix} 8 & 2 & 1 \\ 1 & 7 & 3 \\ 1 & 1 & 6 \end{bmatrix} \begin{bmatrix} 9 \\ 7 \\ 4 \end{bmatrix} = \begin{bmatrix} 90 \\ 70 \\ 40 \end{bmatrix},$$

entonces 10 es un valor propio de A y $[9 \quad 7 \quad 4]^{\mathrm{T}}$ es un vector propio de A asociado a 10. El vector columna $[0.7448453 \quad 0.5793241 \quad 0.3310424]^{\mathrm{T}}$ es un vector propio normalizado asociado a 10.

Con frecuencia se utiliza otra caracterización de los valores propios.

$$\begin{aligned} Av &= \lambda v \\ Av - \lambda v &= 0 \\ Av - \lambda I v &= 0 \\ (A - \lambda I)v &= 0 \end{aligned}$$

Como $v \neq 0$ es solución de un sistema homogéneo, entonces A no puede ser invertible, es decir

$$\det(A - \lambda I) = 0. \tag{9.1}$$

Como A es real, se puede demostrar que $p(\lambda) = \det(A - \lambda I)$ es un polinomio real de grado n:

$$p(\lambda) = p_A(\lambda) = \alpha_0 + \alpha_1\lambda + \alpha_2\lambda^2 + \cdots + \alpha_n\lambda^n. \tag{9.2}$$

Este polinomio se llama el *polinomio característico* de A (algunas veces se considera que el polinomio característico es $\det(\lambda I - A)$). Entonces, **para matrices pequeñas**, se pueden calcular los valores propios, obteniendo las raíces del polinomio característico de A.

Ejemplo 9.2.

$$\begin{aligned} A &= \begin{bmatrix} 1 & -2 \\ 3 & 4 \end{bmatrix} \\ \det(A - \lambda I) &= \det \begin{bmatrix} 1-\lambda & -2 \\ 3 & 4-\lambda \end{bmatrix} \\ &= (1-\lambda)(4-\lambda) + 8 \\ &= \lambda^2 - 5\lambda + 10 \\ \lambda_1 &= 2.5 + i\sqrt{15}/2 \\ \lambda_2 &= 2.5 - i\sqrt{15}/2 \end{aligned}$$

Denotaremos con $\text{espec}(A)$ el conjunto de valores propios de A y λ o λ_i será un valor propio cualquiera. Para el complejo $z = a + ib$, el módulo, norma o tamaño será

$$|z| = \sqrt{x^2 + y^2},$$

que coincide con valor absoluto para números reales.

A continuación algunas definiciones y resultados sobre valores propios. Estos resultados no están necesariamente en el orden conceptual ni en orden para una posible demostración. Además uno puede ser corolario de otro o pueden corresponder al mismo concepto dicho de otra forma.

a. Dos matrices A y B son *semejantes* si existe una matriz C tal que

$$A = C^{-1}BC.$$

b. A es *ortogonal* si $A^{-1} = A^{\mathrm{T}}$.

c. Se denota con A^* o A^H la matriz *transjugada* de A, es decir,

$$A^* = (\bar{A})^{\mathrm{T}} = \overline{A^{\mathrm{T}}}.$$

d. A es *hermitiana* (o hermítica) si $A = A^*$.

e. A es *unitaria* si $A^{-1} = A^*$.

f. Una matriz A es *diagonalizable* si es semejante una matriz diagonal, es decir, existe B invertible y D diagonal tales que

$$D = B^{-1}AB.$$

Resultados:

1.
$$\begin{aligned} \alpha_n &= (-1)^n \\ \alpha_{n-1} &= (-1)^{n-1}\mathrm{traza}(A) \\ \alpha_0 &= \det(A). \end{aligned}$$

2. $\displaystyle\sum_{i=1}^{n} \lambda_i = \mathrm{traza}(A)$

3. $\displaystyle\prod_{i=1}^{n} \lambda_i = \det(A)$

4. Hay n valores propios, reales o complejos, y pueden estar repetidos.

5. Si n es impar, hay por lo menos un valor propio real.

6. El número de valores propios estrictamente complejos (no reales) es par.

7. Sean λ_1, λ_2, ..., λ_k valores propios distintos de A y v^1, v^2, ..., v^k vectores propios asociados correspondientes, entonces estos vectores propios son linealmente independientes.

8. Teorema de Cayley-Hamilton. Si p es el polinomio característico de A, entonces $p(A) = 0$.

9. Si A y B son semejantes, $A = C^{-1}BC$, entonces espec(A) = espec(B).

10. Teorema de Schur. Toda matriz A es semejante a una matriz triangular superior (cuyos elementos diagonales son los valores propios de A). Dicho de otra forma, existe U invertible y T triangular superior tales que $T = U^{-1}AU$. Esta matriz U es unitaria.

11. Si U es unitaria, en particular ortogonal, $||Ux||_2 = ||x||_2$. Así, se dice que las matrices unitarias conservan la norma euclidiana.

12. Si A es simétrica, todos los valores propios son reales.

13. Si A es diagonal, triangular superior o triangular inferior, entonces los valores propios son los elementos diagonales.

14. Teorema espectral. Si A es simétrica, entonces xisten vectores propios v^1, v^2, ..., v^n ortonormales. Si $Q = \begin{bmatrix} v^1 & v^2 & \dots & v^n \end{bmatrix}$, entonces Q es ortogonal y $Q^{\mathrm{T}}AQ$ es una matriz diagonal (con los valores propios de A en la diagonal).

15. Sea A simétrica. La matriz es definida positiva sssi los valores propios son positivos.

16. Sea A simétrica. La matriz es semidefinida positiva sssi los valores propios son no negativos.

17. Si A no es invertible, $\lambda = 0$ es un valor propio.

18. Si A es invertible, $\lambda \in$ espec(A) sssi $1/\lambda \in$ espec(A^{-1}).

19. $\lambda \in$ espec(A) sssi $\lambda - t \in$ espec$(A - tI)$.

20. Para cualquier norma matricial generada $||\ \ ||$,

$$|\lambda| \leq ||A||.$$

21. Si A es ortogonal, $|\lambda| = 1$, para cualquier valor propio real o complejo.

9.1.1 En Scilab

Los valores propios se calculan por medio de la función `spec` (en Matlab se usa `eig`). Si se ha definido una matriz cuadrada `a`, entonces la orden

```
spec(a)
```

da como resultado un vector columna con los n valores propios. La orden

```
[V, L] = spec(a)
```

produce una matriz `L` diagonal, cuyos elementos diagonales son los valores propios y una matriz `V` cuyas columnas son vectores propios normalizados asociados correspondientes.

9.2 Método de la potencia

Este método se puede aplicar para hallar λ_1, el valor propio dominante de una matriz diagonalizable A, cuando éste existe, o sea, si

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|.$$

Una primera versión del método de la potencia es muy sencilla. Dado un x^0 inicial

$$x^{k+1} = Ax^k, \quad k = 0, 1, 2, ... \tag{9.3}$$

Sea $\{v^1, v^2, ..., v^n\}$ una base formada por vectores propios asociados a los valores propios $\lambda_1, \lambda_2, ..., \lambda_n$ respectivamente. Entonces $x^0 \neq 0$ se puede

expresar como combinación de los vectores propios

$$\begin{aligned}
x^0 &= \alpha_1 v^1 + \alpha_2 v^2 + ... + \alpha_n v^n \\
x^1 &= Ax^0 \\
x^1 &= A(\alpha_1 v^1 + \alpha_2 v^2 + ... + \alpha_n v^n) \\
x^1 &= \alpha_1 Av^1 + \alpha_2 Av^2 + ... + \alpha_n Av^n \\
x^1 &= \alpha_1 \lambda_1 v^1 + \alpha_2 \lambda_2 v^2 + ... + \alpha_n \lambda_n v^n \\
x^2 &= Ax^1 \\
&= A(\alpha_1 \lambda_1 v^1 + \alpha_2 \lambda_2 v^2 + ... + \alpha_n \lambda_n v^n) \\
x^2 &= \alpha_1 \lambda_1 Av^1 + \alpha_2 \lambda_2 Av^2 + ... + \alpha_n \lambda_n Av^n \\
x^2 &= \alpha_1 \lambda_1^2 v^1 + \alpha_2 \lambda_2^2 v^2 + ... + \alpha_n \lambda_n^2 v^n \\
&\vdots \\
x^k &= \alpha_1 \lambda_1^k v^1 + \alpha_2 \lambda_2^k v^2 + ... + \alpha_n \lambda_n^k v^n \\
x^k &= \alpha_1 \lambda_1^k \left(v^1 + \sum_{i=2}^{n} \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k v^i \right)
\end{aligned}$$

Esta última factorización está bien definida si $\alpha_1 \neq 0$, o sea, si x^0 no es ortogonal a v^1. Como $|\lambda_i/\lambda_1| < 1$, entonces para valores grandes de k

$$x^k \approx \alpha_1 \lambda_1^k v^1.$$

De manera análoga

$$x^{k+1} \approx \alpha_1 \lambda_1^{k+1} v^1.$$

Entonces

$$x^{k+1} \approx \lambda_1 x^k.$$

Al tomar x_j^k, una componente no nula de x^k,

$$\frac{x_j^{k+1}}{x_j^k} \approx \lambda_1.$$

Ejemplo 9.3. Partiendo de $x^0 = (1, 1, 1)$, hallar el valor propio dominante de

$$A = \begin{bmatrix} -1 & -2 & -3 \\ -4 & -5 & -6 \\ -7 & -8 & -8 \end{bmatrix}.$$

A continuación están los valores k, x_1^k, x_2^k, x_3^k, x_1^k/x_1^{k-1}:

1	-6.000000	-15.000000	-23.000000	-6.00000000
2	105.000000	237.000000	346.000000	-17.50000000
3	-1617.000000	-3681.000000	-5399.000000	-15.40000000
4	25176.000000	57267.000000	83959.000000	-15.56957328
5	-3.915870e+05	-8.907930e+05	-1.306040e+06	-15.55397998
6	6.091293e+06	1.385655e+07	2.031577e+07	-15.55540148
7	-9.475172e+07	-2.155426e+08	-3.160177e+08	-15.55527176
8	1.473890e+09	3.352826e+09	4.915744e+09	-15.55528360
9	-2.292677e+10	-5.215415e+10	-7.646579e+10	-15.55528252
10	3.566324e+11	8.112726e+11	1.189447e+12	-15.55528262
11	-5.547518e+12	-1.261957e+13	-1.850218e+13	-15.55528261
12	8.629321e+13	1.963010e+14	2.878067e+14	-15.55528261

El mecanismo anterior puede conducir hasta una buena aproximación de λ_1, pero tiene un inconveniente: $||x^k|| \to \infty$. La solución es normalizar. Sea $z^0 = x^0$.

$$z^k = Ax^{k-1}, \quad k = 1, 2, 3, ... \tag{9.4}$$

$$x^k = \frac{z^k}{||z^k||_2}. \tag{9.5}$$

Ejemplo 9.4. Usar las fórmulas anteriores, partiendo de $x^0 = (1, 1, 1)$, para hallar el valor propio dominante de

$$A = \begin{bmatrix} -1 & -2 & -3 \\ -4 & -5 & -6 \\ -7 & -8 & -8 \end{bmatrix}.$$

A continuación están los valores k, x_1^k, x_2^k, x_3^k, z_1^k/x_1^{k-1}:

1	-0.213470	-0.533676	-0.818303	-6.00000000

```
 2       0.242870       0.548191       0.800313 -17.50000000
 3      -0.240212      -0.546829      -0.802045 -15.40000000
 4       0.240454       0.546954       0.801887 -15.56957328
 5      -0.240432      -0.546942      -0.801902 -15.55397998
 6       0.240434       0.546943       0.801900 -15.55540148
 7      -0.240434      -0.546943      -0.801901 -15.55527176
 8       0.240434       0.546943       0.801901 -15.55528360
 9      -0.240434      -0.546943      -0.801901 -15.55528252
10       0.240434       0.546943       0.801901 -15.55528262
11      -0.240434      -0.546943      -0.801901 -15.55528261
12       0.240434       0.546943       0.801901 -15.55528261
```

El siguiente esquema, además de incluir la normalización, tiene una manera más eficiente de aproximar λ.

Algoritmo de la potencia

para $k = 1, ...,$`maxit`
$z^k = Ax^{k-1}$
$x^k = \dfrac{z^k}{||z^k||_2}$
$\lambda_1^k = x^{k^{\mathrm{T}}} z^k$
si $|\lambda_1^k - \lambda_1^{k-1}| \leq \varepsilon$, **parar**
fin-para

El proceso se detiene satisfactoriamente cuando dos aproximaciones, λ_1^k y λ_1^{k-1}, son muy parecidas. La salida no deseada se tiene cuando se llega al número máximo de iteraciones.

La rapidez de la convergencia está ligada al valor $|\lambda_1/\lambda_2|$. Si este valor es cercano a 1, la convergencia es lenta. Si es mucho mayor que 1, la convergencia es rápida.

Ejemplo 9.5. Hallar el valor propio dominante de

$$A = \begin{bmatrix} -1 & -2 & -3 \\ -4 & -5 & -6 \\ -7 & -8 & -8 \end{bmatrix}$$

partiendo de $x^0 = (1, 1, 1)$.

Los siguientes valores corresponden a k, z_1^k, z_2^k, z_3^k, x_1^k, x_2^k, x_3^k, λ_1^k:

```
 1      -6.000000     -15.000000     -23.000000
        -0.213470      -0.533676      -0.818303  28.10693865
 2       3.735732       8.432082      12.310128
         0.242870       0.548191       0.800313  15.38164285
 3      -3.740191      -8.514312     -12.488120
        -0.240212      -0.546829      -0.802045  15.57034584
 4       3.740005       8.507264      12.472478
         0.240454       0.546954       0.801887  15.55390218
 5      -3.740024      -8.507910     -12.473909
        -0.240432      -0.546942      -0.801902  15.55540852
 6       3.740022       8.507851      12.473779
         0.240434       0.546943       0.801900  15.55527112
 7      -3.740022      -8.507857     -12.473791
        -0.240434      -0.546943      -0.801901  15.55528366
 8       3.740022       8.507856      12.473790
         0.240434       0.546943       0.801901  15.55528251
 9      -3.740022      -8.507856     -12.473790
        -0.240434      -0.546943      -0.801901  15.55528262
10       3.740022       8.507856      12.473790
         0.240434       0.546943       0.801901  15.55528261
11      -3.740022      -8.507856     -12.473790
        -0.240434      -0.546943      -0.801901  15.55528261
```

El último x^k obtenido es una buena aproximación de un vector propio normalizado asociado a λ_1. $\diamond$

9.3 Método de la potencia inversa

Este método se puede aplicar para hallar λ_n, el valor propio menos dominante de una matriz diagonalizable e invertible A, cuando éste existe, o sea, si

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \cdots > |\lambda_n| > 0.$$

Si A es invertible y tiene valores propios λ_1, λ_2,..., λ_n, entonces los valores propios de A^{-1} son

$$\frac{1}{\lambda_1}, \ \frac{1}{\lambda_2}, ..., \frac{1}{\lambda_n}.$$

El valor propio dominante de A^{-1} es justamente $1/\lambda_n$. Entonces se puede aplicar el método de la potencia a A^{-1}. En lugar de escribir explícitamente

$z^k = A^{-1}x^{k-1}$ es preferible presentarlo como la solución del sistema $Az^k = x^{k-1}$.

Potencia inversa

para $k = 1, ...,$`maxit`

resolver $Az^k = x^{k-1}$

$$x^k = \frac{z^k}{||z^k||_2}$$

$$\sigma_1^k = x^{k^{\mathrm{T}}} z^k$$

si $|\sigma_1^k - \sigma_1^{k-1}| \leq \varepsilon$, **parar**

fin-para

Cuando se obtenga la convergencia, $\lambda_n \approx 1/\sigma_1^k$.

Ejemplo 9.6. Aplicar, partiendo de $x^0 = (1, 1, 1)$, el método de la potencia inversa para obtener el valor propio λ_n de la matriz

$$A = \begin{bmatrix} -1 & -2 & -3 \\ -4 & -5 & -6 \\ -7 & -8 & -8 \end{bmatrix}.$$

```
1       1.000000      -1.000000       0.000000
        0.707107      -0.707107       0.000000   1.41421356
2       3.771236      -5.421152       2.121320
        0.543702      -0.781572       0.305832   6.93621735
3       3.839896      -5.810817       2.412678
        0.520948      -0.788337       0.327321   7.37098425
4       3.818745      -5.807259       2.424942
        0.518766      -0.788900       0.329422   7.36121039
5       3.816531      -5.806630       2.425988
        0.518557      -0.788954       0.329622   7.35991006
6       3.816317      -5.806567       2.426087
        0.518537      -0.788959       0.329641   7.35978177
7       3.816297      -5.806561       2.426096
        0.518535      -0.788960       0.329643   7.35976946
8       3.816295      -5.806560       2.426097
        0.518535      -0.788960       0.329643   7.35976828
```

```
 9       3.816294      -5.806560      2.426097
         0.518535      -0.788960      0.329643   7.35976817
10       3.816294      -5.806560      2.426097
         0.518535      -0.788960      0.329643   7.35976816
11       3.816294      -5.806560      2.426097
         0.518535      -0.788960      0.329643   7.35976815
12       3.816294      -5.806560      2.426097
         0.518535      -0.788960      0.329643   7.35976815
```

Entonces $\lambda_n \approx 1/7.35976815 = 0.135873845$

9.4 Factorización QR

Sea $A \in \mathbb{R}^{m\times n}$. Una factorización QR de A consiste en encontrar matrices Q y R tales que

- $A = QR$.
- $Q \in \mathbb{R}^{m\times m}$ es ortogonal.
- $R \in \mathbb{R}^{m\times n}$ es triangular superior ($r_{ij} = 0$ si $i > j$).

El proceso de factorización QR, por medio de diferentes clases de matrices ortogonales, va obteniendo ceros en lugares adecuados. Supongamos que por medio de Q_1 ortogonal, la matriz Q_1A tiene ceros en sitios adecuados. Ahora, con Q_2 ortogonal, se busca que al hacer el producto Q_2Q_1A haya ceros en otros sitios, sin perder los que ya tenía Q_1A. Finalmente se obtiene

$$Q_rQ_{r-1}\cdots Q_2Q_1A = R \text{ triangular superior.}$$

Como las matrices Q son ortogonales, entonces

$$\begin{aligned} Q_1^{\mathrm{T}}Q_2^{\mathrm{T}}\cdots Q_{r-1}^{\mathrm{T}}Q_r^{\mathrm{T}}Q_rQ_{r-1}\cdots Q_2Q_1A &= Q_1^{\mathrm{T}}Q_2^{\mathrm{T}}\cdots Q_{r-1}^{\mathrm{T}}Q_r^{\mathrm{T}}R \\ A &= \underbrace{Q_1^{\mathrm{T}}Q_2^{\mathrm{T}}\cdots Q_{r-1}^{\mathrm{T}}Q_r^{\mathrm{T}}}R \\ A &= QR \end{aligned}$$

En los programas, generalmente se empieza con A y sobre ella se va reescribiendo el producto Q_1A, después Q_2Q_1A. Al final se tendrá, en donde

estaba A, la matriz R. Por otro lado, se puede empezar con $Q = I$, y encima se va reescribiendo el producto IQ_1^{T}, después $Q_1^{\mathrm{T}}Q_2^{\mathrm{T}}$. Finalmente en Q se tendrá el producto $Q_1^{\mathrm{T}}Q_2^{\mathrm{T}}\cdots Q_{r-1}^{\mathrm{T}}Q_r^{\mathrm{T}}$.

9.4.1 Matrices de Householder

Sea $v \in \mathbb{R}^{n\times 1}$, $v \neq 0$, $u = v/||v||$ (vector columna de norma 1). Una matriz de Householder es una matriz de la forma

$$H = H_v = H(v) = I_n - \frac{2}{v^{\mathrm{T}}v} v\, v^{\mathrm{T}} = I_n - 2uu^{\mathrm{T}}.$$

A veces, al mismo tiempo que se obtiene el vector v deseado, se calcula el número

$$\beta = \frac{2}{v^{\mathrm{T}}v} \ ,$$

entonces es común expresar H en función de v y de β, aunque β no es ncesario. Simplemente, desde el punto de vista de eficiencia, si se conoce β no es interesante volverlo a calcular (son $2n-1$ "flops").

$$H = H(v,\beta) = I_n - \beta v\, v^{\mathrm{T}} \ .$$

La matriz H tiene dos características importantes, es *simétrica y ortogonal.*

Ademas, si $x \in \mathbb{R}^{n\times 1}$ se puede escoger v para que

$$H_v x \in < e^1 > \ .$$

En Álgebra Lineal, dados x^1, x^2, ... x^k vectores del espacio vectorial en consideración, $< x^1, x^2, ..., x^k >$ denota el subespacio generado por estos vectores, es decir, el conjunto de todas las combinaciones lineales de estos vectores:

$$< x^1, x^2, ..., x^k >= \{\lambda_1 x^1 + \lambda_2 x^2 + \cdots + \lambda_k x^k \ : \ \lambda_i \in \mathbb{R}\}.$$

Entonces,

$$H_v x = \alpha e^1.$$

Sea $U = \{\xi \in \mathbb{R}^{n\times 1} : v^{\mathrm{T}}\xi = 0\}$, o sea, el hiperplano perpendicular a v y que pasa por el origen. Dicho de otra forma, U es el complemento ortogonal del subespacio generado por v, $U = < v >^{\perp}$.

Sea $x \in \mathbb{R}^{n\times 1}$, $y = Hx$, $p = (x+y)/2$, o sea, el punto medio del segmento que une a x con y. Se puede verificar que

$$v^{\mathrm{T}} p = 0\,, \text{ o sea, } \; p \in U\,.$$

Si

$$z = x - p$$

se puede verificar que

$$p^{\mathrm{T}} z = 0.$$

Como $p + z = x$, entonces se deduce que p es la proyección de x sobre U, y como p es el punto medio entre x y y, entonces y es el punto simétrico de x con respecto a hiperplano U o la reflexión de x con respecto a U.

Como H es ortogonal, entonces $||y|| = ||x||$. **Si se desea que $y = \alpha e^1$,** entonces

$$y = \pm||x||\, e^1\,.$$

Sea $\xi \in U$, o sea, $v^{\mathrm{T}}\xi = 0$. Fácilmente se comprueba que

$$(x-y)^{\mathrm{T}}\xi = 0\,.$$

Si $x = y$, entonces $x = \pm||x||\, e^1$. Basta con tomar $H = I$, y así, $Hx = \lambda e^1$. Si $x \neq y$, se puede tomar

$$v = x \mp ||x|| e^1$$

Ejemplo 9.7.

$$x = \begin{bmatrix} -2 \\ -1 \\ 2 \end{bmatrix}, \quad y = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}, \quad v = \begin{bmatrix} -5 \\ -1 \\ 2 \end{bmatrix}, \quad H = \begin{bmatrix} -2/3 & -1/3 & 2/3 \\ -1/3 & 14/15 & 2/15 \\ 2/3 & 2/15 & 11/15 \end{bmatrix}$$

O también,

$$x = \begin{bmatrix} -2 \\ -1 \\ 2 \end{bmatrix}, \quad y = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}, \quad v = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, \quad H = \begin{bmatrix} 2/3 & 1/3 & -2/3 \\ 1/3 & 2/3 & 2/3 \\ -2/3 & 2/3 & -1/3 \end{bmatrix}$$

Es usual escoger v tal que $v_1 = 1$, así sólo se requiere almacenar los valores v_2, v_3, ..., v_n. Generalmente estos valores se pueden almacenar donde estaban x_2, x_3, x_n. Además, no es necesario construir explícitamente H, basta con conocer v y β.

Denotaremos por $\overline{H}(x)$ la matriz que proyecta x sobre el subespacio $< e^1 >$.

La siguiente función, ver [Par80] y [GoVa96], presenta una manera eficiente de calcular v y β a partir de un vector columna x. Está escrita en seudocódigo utilizando parcialmente notación de Scilab.

$[v,\ \beta] = \texttt{vHouse}(x)$
  $n = \texttt{dim}\,(x)$
  $t = x(2{:}n)^{\mathrm{T}} x(2{:}n)$
  $v = [\,1\,;\ x(2{:}n)\,]$
  si $t = 0$
    $\beta = 0$
  sino
    $\nu = \sqrt{x_1^2 + t}$
    si $x_1 \leq 0$
      $v_1 = x_1 - \nu$
    sino
      $v_1 = -t/(x_1 + \nu)$
    fin-si
    $\beta = 2v_1^2/(t + v_1^2)$
    $v = v/v_1$
  fin-si
fin `vHouse`

En resumen, dado $x = \in \mathbb{R}^n$,

$$[v,\ \beta] = \text{ vHouse}(x)$$
$$\overline{H}(x) = H(v, \beta) = I - \beta v v^{\mathrm{T}}\,.$$

Ejemplo 9.8.

$$x = \begin{bmatrix} -2 \\ -1 \\ 2 \end{bmatrix}, \quad v = \begin{bmatrix} 1 \\ 1/5 \\ -2/5 \end{bmatrix}, \quad \beta = \frac{5}{3}\,.$$

9.4.2 Matrices de Givens

Esta es otra clase de matrices ortogonales. Sea θ un ángulo y

$$\begin{aligned} c &= \cos(\theta) \\ s &= \operatorname{sen}(\theta), \end{aligned}$$

La matriz de Givens, en $\mathbb{R}^{n\times n}$, es simplemente un rotación definida en el plano de las variables i y k:

$$G = G(i,k,c,s,n) = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & \ddots & & & & & \\ 0 & 0 & \cdots & c & & s & & 0 \\ \vdots & & & & \ddots & & & \\ 0 & 0 & \cdots & -s & & c & & 0 \\ \vdots & & & & & & \ddots & \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \begin{matrix} \\ \\ \\ i \\ \\ k \\ \\ \\ \end{matrix}$$

El producto $y = G^{\mathrm{T}}x$ se calcula muy fácilmente:

$$y_j = \begin{cases} cx_i - sx_k & \text{si } \ j = i, \\ sx_i + cx_k & \text{si } \ j = k, \\ x_j & \text{en los demás casos.} \end{cases}$$

Si se desea que $y_k = 0$, basta con tomar

$$\begin{aligned} c &= \frac{x_i}{\sqrt{x_i^2 + x_k^2}}\ , \\ s &= \frac{-x_k}{\sqrt{x_i^2 + x_k^2}}\ . \end{aligned}$$

En la práctica, es mejor utilizar la siguiente versión para el cálculo de c y s (ver [GoVa96]),

```
[c, s] = csGivens(a, b)
    si b = 0
        c = 1
        s = 0
    sino
        si |b| > |a|
            t = −a/b
            s = 1/√(1 + t²)
            c = st
        sino
            t = −b/a
            c = 1/√(1 + t²)
            s = ct
        fin-si
    fin-si
fin csGivens
```

Por medio de esta función

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix} .$$

Ejemplo 9.9. Para el vector

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \end{bmatrix} \quad \text{por medio de la función se obtiene} \quad \begin{array}{l} c = 0.5547002 \\ s = 0.8320503 \end{array}$$

y así

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 2 \\ -3 \end{bmatrix} = \begin{bmatrix} 3.6055513 \\ 0 \end{bmatrix} .$$

9.4.3 Factorización QR con matrices de Householder

Para facilitar la presentación del algoritmo, usaremos la siguiente notación. Si $H \in \mathbb{R}^{p\times p}$ es una matriz de Householder, con $p \leq n$,

$$\widehat{H} = \widehat{H}(n, H) = \begin{cases} H & \text{si } p = n \\ \begin{bmatrix} I_{n-p} & 0 \\ 0 & H \end{bmatrix} & \text{si } p < n \end{cases}$$

La matriz $\widehat{H} \in \mathbb{R}^{n\times n}$ también es ortogonal.

En lo que sigue se supondrá que A siempre indica la matriz obtenida al hacer los productos efectuados hasta este momento, o sea, $Q_k \cdots Q_2 Q_1 A_{\text{inicial}}$.

Inicialmente se supone que $Q = I_m$ y se buscan ceros por debajo de a_{11}, o sea, se construye $Q_1 = H_1 \in \mathbb{R}^{m\times m}$ tal que $H_1 A(:,1) = \alpha_1 e^1 \in \mathbb{R}^{m\times 1}$:

$$\begin{aligned} [v,\beta] &= \text{vHouse}(A(1\!:\!m,1)) \\ H_1 &= H(v,\beta) \\ A &= H_1 A \\ Q &= Q H_1^{\mathrm{T}} = Q H_1. \end{aligned}$$

En seguida, se trabaja únicamente con las filas 2,..., m de A. Se construye $H_2 \in \mathbb{R}^{(m-1)\times(m-1)}$ tal que $H_2\, A(2\!:\!m,2) = \alpha_2 e^1 \in \mathbb{R}^{(m-1)\times 1}$, o sea,

$$\begin{aligned} [v,\beta] &= \text{vHouse}(A(2\!:\!m,2)) \\ H_2 &= H(v,\beta) \\ \widehat{H}_2 &= \widehat{H}(m,H_2) \\ A &= \widehat{H}_2 A \\ Q &= Q\widehat{H}_2 \end{aligned}$$

En general,

$$\begin{aligned} [v,\beta] &= \text{vHouse}(A(k\!:\!m,k)) \\ H_k &= H(v,\beta) \\ \widehat{H}_k &= \widehat{H}(m,H_k) \\ A &= \widehat{H}_k A \\ Q &= Q\widehat{H}_k \end{aligned}$$

Como se supone que en la iteración k, las columnas 1, ..., $k-1$ de A son nulas debajo de la diagonal, entonces no es necesario recalcularlas. La presentación

formal anterior es exacta pero ineficiente, es mejor

$$\begin{aligned} [v, \beta] &= \text{vHouse}(A(k\!:\!m, k)) \\ H_k &= H(v, \beta) \\ A(k\!:\!m, :) &= H_k A(k\!:\!m, :) \\ Q(:, k\!:\!m) &= Q(:, k\!:\!m) H_k \end{aligned}$$

A continuación hay dos presentaciones de la factorización QR por medio de matrices de Householder, la primera versión es más fácil de presentar.

$[Q, R] =$ `QR_House` (A)
  $[m, n] = \text{tamaño}(A)$
  $Q = I_m$
  para $k = 1 : \min(m, n)$
    $[v, \beta] = \text{vHouse}(A(k\!:\!m, k)$
    $H = H(v, \beta)$
    $\widehat{H} = \widehat{H}(m, H)$
    $A = \widehat{H}\ A$
    $Q = Q\widehat{H}$
  fin-para
  $R = A$
fin `QR_House`

Esta segunda versión es mucho más eficiente.

$[Q, R] =$ `QR_House` (A)
  $[m, n] = \text{tamaño}(A)$
  $Q = I_m$
  para $k = 1 : \min(m, n)$
    $[v, \beta] = \text{vHouse}(A(k\!:\!m, k)$
    $H = H(v, \beta)$
    $A(k\!:\!m, k\!:\!n) = H\ A(k\!:\!m, k\!:\!n)$
    $Q(:, k\!:\!m) = Q(:, k\!:\!m)\, H$
  fin-para
  $R = A$
fin `QR_House`

Ejemplo 9.10. Obtener la factorización QR de

$$A = \begin{bmatrix} 2 & 3 & 4 \\ 5 & 4 & 3 \\ 2 & 1 & 0 \\ -1 & -2 & -3 \\ -4 & -5 & -4 \end{bmatrix}$$

utizando matrices de Householder.

```
k = 1
beta = 0.717157
v : 1   -0.98598563  -0.39439425   0.19719713   0.78878851

 H  =
    0.2828427    0.7071068    0.2828427   -0.1414214   -0.5656854
    0.7071068    0.3028029   -0.2788789    0.1394394    0.5577577
    0.2828427   -0.2788789    0.8884485    0.0557758    0.2231031
   -0.1414214    0.1394394    0.0557758    0.9721121   -0.1115515
   -0.5656854    0.5577577    0.2231031   -0.1115515    0.5537938

A  =
    7.0710678    7.0710678    5.939697
    0           -0.0140144    1.0874867
    0           -0.6056057   -0.7650053
    0           -1.1971971   -2.6174973
    0           -1.7887885   -2.4699893

 Q  =
    0.2828427    0.7071068    0.2828427   -0.1414214   -0.5656854
    0.7071068    0.3028029   -0.2788789    0.1394394    0.5577577
    0.2828427   -0.2788789    0.8884485    0.0557758    0.2231031
   -0.1414214    0.1394394    0.0557758    0.9721121   -0.1115515
   -0.5656854    0.5577577    0.2231031   -0.1115515    0.5537938
-----------------------------------------------------------------
k = 2
beta = 1.006267
v : 1    0.26914826   0.53206814   0.79498802

 H  =
```

```
   -0.0062674   -0.2708351   -0.5354028   -0.7999705
   -0.2708351    0.9271052   -0.1441027   -0.2153107
   -0.5354028   -0.1441027    0.7151292   -0.4256388
   -0.7999705   -0.2153107   -0.4256388    0.3640330

 A  =
    7.0710678    7.0710678    5.939697
    0            2.236068     3.5777088
    0            0           -0.0947664
    0            0           -1.2925295
    0            0           -0.4902926

 Q  =
    0.2828427    0.4472136    0.2128929   -0.2797022   -0.7722974
    0.7071068   -0.4472136   -0.4807445   -0.2596204   -0.0384964
    0.2828427   -0.4472136    0.8431415   -0.0337898    0.0892790
   -0.1414214   -0.4472136   -0.1021209    0.6599727   -0.5779337
   -0.5656854   -0.4472136   -0.0473832   -0.6462647   -0.2451463
-----------------------------------------------------------------
k=3
beta = 1.068392
v : 1    0.87309062   0.33118770

 H  =
   -0.0683918   -0.9328028   -0.3538382
   -0.9328028    0.1855786   -0.3089328
   -0.3538382   -0.3089328    0.8828131

 A  =
    7.0710678    7.0710678    5.939697
    0            2.236068     3.5777088
    0            0            1.3856406
    0            0            0
    0            0            0

Q  =
    0.2828427    0.4472136    0.5196152   -0.0119059   -0.6707147
    0.7071068   -0.4472136    0.2886751    0.4121526    0.2163259
    0.2828427   -0.4472136   -0.0577350   -0.8203366   -0.2090802
   -0.1414214   -0.4472136   -0.4041452    0.3962781   -0.6779604
```

```
-0.5656854   -0.4472136    0.6928203    0            0
```

Observaciones:

- No es necesario calcular explícitamente las matrices H (en el ejemplo anterior aparecen, pero simplemente de manera ilustrativa). Basta con conocer β y v.
- Es necesario implementar eficientemente el producto $H_k A(k:m,:)$ a partir de la información: $A(k:m,:)$, β y v.
- De manera análoga, es necesario implementar eficientemente el producto $Q(:,k:m)H_k$ a partir de la información: $Q(:,k:m)$, β y v.

9.4.4 Factorización QR con matrices de Givens

Al utilizar matrices ortogonales de Givens, también se busca, columna por columna, anular los elementos debajo de la diagonal. Con matrices de Householder, se utilizaba una matriz para cada columna. Con matrices de Givens, en la columna k, se utiliza una matriz para anular $a_{m,k}$, después otra matriz para anular $a_{m-1,k}$, después otra matriz para anular $a_{m-2,k}$ y , finalmente, otra matriz para anular $a_{k+1,k}$.

$[Q,R] =$ `QR_Givens`(A)
  $[m,n] =$ tamaño(A)
  $Q = I_m$
  para $k = 1 : \min(m,n)$
    para $i = m : -1 : k+1$
      $[c,s] =$ csGivens$(a_{i-1,k}\ ,\ a_{ik})$
      $G = G(i-1,i,c,s,m)$
      $A = G^{\mathrm{T}}A$
      $Q = QG$
    fin-para
  fin-para
  $R = A$
fin `QR_Givens`

Ejemplo 9.11. Obtener la factorización QR de

$$A = \begin{bmatrix} 2 & 3 & 4 \\ 5 & 4 & 3 \\ 2 & 1 & 0 \\ -1 & -2 & -3 \\ -4 & -5 & -4 \end{bmatrix}$$

utizando matrices de Givens.

```
k = 1
i = 5
c = -0.242536    s = 0.970143
 G  =
    1     0     0     0            0
    0     1     0     0            0
    0     0     1     0            0
    0     0     0    -0.2425356    0.9701425
    0     0     0    -0.9701425   -0.2425356
 A  =
    2            3           4
    5            4           3
    2            1           0
    4.1231056    5.3357838   4.6081769
    0           -0.7276069  -1.940285
 Q  =
    1     0     0     0            0
    0     1     0     0            0
    0     0     1     0            0
    0     0     0    -0.2425356    0.9701425
    0     0     0    -0.9701425   -0.2425356

i = 4
c = -0.436436    s = 0.899735
 G  =
    1     0     0            0           0
    0     1     0            0           0
    0     0    -0.4364358    0.8997354   0
    0     0    -0.8997354   -0.4364358   0
    0     0     0            0           1
```

```
 A  =
    2            3            4
    5            4            3
   -4.5825757   -5.2372294   -4.1461399
    0           -1.4289915   -2.0111733
    0           -0.7276069   -1.940285
 Q  =
    1     0     0            0            0
    0     1     0            0            0
    0     0    -0.4364358    0.8997354    0
    0     0     0.2182179    0.1058512    0.9701425
    0     0     0.8728716    0.4234049   -0.2425356
...
k = 3
...
i = 4
c = -0.612372    s = 0.790569
 A  =
   -7.0710678   -7.0710678   -5.939697
    0           -2.236068    -3.5777088
    0            0           -1.3856406
    0            0            0
    0            0            0
 Q  =
   -0.2828427   -0.4472136   -0.5196152    0.6708204    0
   -0.7071068    0.4472136   -0.2886751   -0.2236068    0.4082483
   -0.2828427    0.4472136    0.0577350    0.2236068   -0.8164966
    0.1414214    0.4472136    0.4041452    0.6708204    0.4082483
    0.5656854    0.4472136   -0.6928203    0            0
```

Para que la (o una) factorización QR de A sea eficiente hay que tener en cuenta, entre otros, los siguietes detalles:

- No es necesario calcular explícitamente las matrices G (en el ejemplo anterior aparecen, pero simplemente de manera ilustrativa). Basta con conocer c, s, e i. Obsérvese que siempre se trata de las filas $i-1$ e i.

- Es necesario implementar eficientemente el producto $G^{\mathrm{T}}A$ a partir de la información: A, c y s.

- De manera análoga, es necesario implementar eficientemente el producto QG a partir de la información: Q, c y s.

En general para efectuar, sobre B, el producto $G(i,j,c,s,m)^{\mathrm{T}}B$ basta con hacer:

$$\begin{aligned} x &= B(i,:) \\ B(i,:) &= c\,B(i,:) - s\,B(j,:) \\ B(j,:) &= s\,x + c\,B(j,:) \end{aligned}$$

En el proceso de factorización QR, si se está buscando un cero en la posición (i,k) de la matriz A, se modifican únicamente, la filas $i-1$ e i, pero se debe tener en cuenta que las columnas 1, ..., $k-1$ son nulas por debajo de la diagonal. Entonces se reduce el número de operaciones.

$$\begin{aligned} a_{i-1,k} &= c\,a_{i-1,k} - s\,a_{ik} \\ a_{ik} &= 0 \\ t &= A(i-1,k+1:n) \\ A(i-1,k+1:n) &= c\,t - s\,A(i,k+1:n) \\ A(i,k+1:n) &= s\,t + c\,A(i,k+1:n) \end{aligned}$$

En general para efectuar, sobre B, el producto $B\,G(i,j,c,s,m)$ basta con hacer:

$$\begin{aligned} x &= B(:,i) \\ B(:,i) &= c\,B(:,i) + s\,B(:,j) \\ B(:,j) &= -s\,x + c\,B(:,j) \end{aligned}$$

9.4.5 Solución por mínimos cuadrados

Una de las aplicaciones importantes de la factorización QR es la solución de sistemas de ecuaciones lineales por mínimos cuadrados. El método más

popular para mínimos cuadrados es el de las ecuaciones normales. Sin embargo, en los casos, cuando el condicionamiento de A es muy grande comparado comparado con el residuo mínimo [GoVa96], el método QR resulta más preciso y estable.

Una propiedad importantísima de las matrices ortogonales es que preservan la norma euclidiana. Si Q es ortogonal, entonces

$$||Qx|| = ||x||.$$

Esto quiere decir que obtener el mínimo de $||Ax-b||_2^2$ es equivalente a buscar el mínimo de $||PAx - Pb||_2^2$ para cualquier matriz ortogonal P. Si $QR = A$ es la factorización QR de A, entonces, se desea minimizar

$$||Q^{\mathrm{T}}Ax - Q^{\mathrm{T}}b||_2^2 = ||Q^{\mathrm{T}}QRx - Q^{\mathrm{T}}b||_2^2 = ||Rx - Q^{\mathrm{T}}b||_2^2.$$

Sea $A \in \mathbb{R}^{m\times n}$, $c = Q^{\mathrm{T}}b$,

$$R = \begin{bmatrix} U \\ 0_{qn} \end{bmatrix}, \quad c = \begin{bmatrix} d \\ r \end{bmatrix},$$

con $U \in \mathbb{R}^{p\times n}$ "triangular" superior, cuya última fila no es nula, $d \in \mathbb{R}^{p\times 1}$, $r \in \mathbb{R}^{q\times 1}$, $p + q = m$. Entonces

$$Rx - c = \begin{bmatrix} Ux - d \\ -r \end{bmatrix}$$
$$||Ax - b||_2^2 = ||Ux - d||_2^2 + ||r||_2^2.$$

Basta con **buscar** x **solución de** $Ux = d$. Si el sistema anterior tiene solución, entonces

$$\min_{x\in\mathbb{R}^n} ||Ax - b||_2^2 = ||r||_2^2.$$

Si U es cuadrada ($\in \mathbb{R}^{n\times n}$) e invertible, la solución es única.

Ejemplo 9.12. Resolver por mínimos cuadrados el sistema $Ax = b$, donde

$$A = \begin{bmatrix} 2 & 3 & 4 \\ 5 & 4 & 3 \\ 2 & 1 & 0 \\ -1 & -2 & -3 \\ -4 & -5 & -4 \end{bmatrix}, \quad b = \begin{bmatrix} 29.1 \\ 33.9 \\ 7.0 \\ -20.1 \\ -38.9 \end{bmatrix}$$

```
Q  =
  -0.2828427   -0.4472136   -0.5196152    0.6708204    0
  -0.7071068    0.4472136   -0.2886751   -0.2236068    0.4082483
  -0.2828427    0.4472136    0.0577350    0.2236068   -0.8164966
   0.1414214    0.4472136    0.4041452    0.6708204    0.4082483
   0.5656854    0.4472136   -0.6928203    0            0

R  =
  -7.0710678   -7.0710678   -5.939697
   0           -2.236068    -3.5777088
   0            0           -1.3856406
   0            0            0
   0            0            0

c :     -59.029274   -21.108482   -5.6753531    0.0223607   -0.0816497

U  =
  -7.0710678   -7.0710678   -5.939697
   0           -2.236068    -3.5777088
   0            0           -1.3856406

d :     -59.029274   -21.108482   -5.6753531

r  :     0.0223607   -0.0816497

x  :    2.0208333    2.8866667    4.0958333
```

Así, $||r||_2^2 = 0.0071667$.

9.5 Método QR para valores propios de matrices simétricas

El método más popular para obtener los valores propios de una matriz simétrica (todos reales) es el método QR. Es posiblemente el más eficente para casos generales. El proceso tiene dos pasos:

1. Obtener, por matrices ortogonales, una matriz T tridiagonal simétrica semejante a A, o sea encontrar Q ortogonal tal que

$$QAQ^{\mathrm{T}} = T \text{ tridiagonal simétrica.}$$

2. Obtener los valores propios de T.

9.5.1 Tridiagonalización por matrices de Householder para matrices simétricas

Sea $A \in \mathbb{R}^{n\times n}$ simétrica, $\widehat{H} = \widehat{H}(n, \overline{H}(A(2:n,1))$. Es claro que $\widehat{H}A$ es nula, en la columna 1, por debajo de la subdiagonal. Se puede observar, y también demostrar, que $\widehat{H}A\widehat{H}$, además de ser nula en la primera columna por debajo de la subdiagonal, también es nula en la primera fila a la derecha de la superdiagonal, y obviamente también es simétrica....

Ejemplo 9.13.

```
A  =
   2     3     4     5
   3    -1     0     1
   4     0    -2     8
   5     1     8    10

H  =
   1     0            0            0
   0     0.4242641    0.5656854    0.7071068
   0     0.5656854    0.4441896   -0.6947630
   0     0.7071068   -0.6947630    0.1315463

H A  =
   2           3           4           5
   7.0710678   0.2828427   4.5254834   12.020815
   0          -1.2604484  -6.4464829  -2.8284271
   0          -0.5755605   2.4418963  -3.5355339

H A H  =
   2           7.0710678   0           0
   7.0710678   11.18      -6.1814444  -1.3628445
   0          -6.1814444  -1.6113918   3.2154369
   0          -1.3628445   3.2154369  -2.5686082
```

Este proceso se realiza en la otras columnas y filas y se obtiene una matriz tridiagonal, simétrica, semejante a A. Como es costumbre, los productos realizados se reescriben sobre A.

$A =$ `triHouse`(A)
  $n =$ `dim` (A)
  para $k = 1 : n - 2$
    $x = A(k + 1 : n, k)$
    $H = \overline{H}(x)$
    $\widehat{H} = \widehat{H}(n, H)$
    $A = \widehat{H} A \widehat{H}$
  fin-para
fin `triHouse`

Ejemplo 9.14.

```
A  =
   2.    3.    4.    5.
   3.  - 1.    0.    1.
   4.    0.  - 2.    8.
   5.    1.    8.    10.

k = 1

H  =
   0.4242641    0.5656854    0.7071068
   0.5656854    0.4441896  - 0.6947630
   0.7071068  - 0.6947630    0.1315463

A  =
   2.           7.0710678    0.           0.
   7.0710678    11.18      - 6.1814444  - 1.3628445
   0.         - 6.1814444  - 1.6113918    3.2154369
   0.         - 1.3628445    3.2154369  - 2.5686082

k = 2

H  =
 - 0.9765473  - 0.2153028
 - 0.2153028    0.9765473
```

```
A  =
   2.           7.0710678    0.           0.
   7.0710678    11.18        6.3298973    0.
   0.           6.3298973  - 0.3036510  - 2.7160739
   0.           0.         - 2.7160739  - 3.876349
```

Tal como está descrito el algoritmo, se supone que se hace explícitamente el producto $\widehat{H}A\widehat{H}$. En realidad se puede hacer de manera más eficiente, teniendo en cuenta que una parte de $\widehat{H}$ es la identidad, que se conoce el nuevo valor de $a_{k+1,k}$, que debajo habrá ceros, y que $\widehat{H}A\widehat{H}$ también es simétrica.

$A =$ `triHouse`(A)
  $n =$ `dim` (A)
  para $k = 1 : n-2$
    $x = A(k+1:n, k)$
    $[v, \beta] =$ vHouse(x)
    $p = \beta\, A(k+1:n, k+1:n)\, v$
    $w = p - (\beta/2)\,(p^{\mathrm{T}}\, v)\, v$
    $a_{k+1,k} = a_{k,k+1} = ||x||$
    $A(k+2:n, k) = 0$
    $A(k, k+2:n) = 0$
    $A(k+1:n, k+1:n) = A(k+1:n, k+1:n) - v\, w^{\mathrm{T}} - w\, v^{\mathrm{T}}$
  fin-para
fin `triHouse`

9.5.2 Tridiagonalización por matrices de Givens para matrices simétricas

Con los conceptos e ideas de la factorización QR por medio de matrices de Givens y de la tridiagonalización con matrices de Householder, resulta naturalmente el proceso de tridiagonalización con matrices de Givens.

Primero se busca "tridiagonalizar" la primera columna y primera fila, o sea, se buscan ceros por debajo de la subdiagonal y a la derecha de la superdiagonal. Para ello se busca un cero en la posición $(n, 1)$, después en la posición $(n-1, 1)$, así sucesivamente hasta la posición $(3, 1)$. Al mismo tiempo se hace lo análogo con la primera fila.

Después se trabaja con segunda columna y segunda fila, y así sucesivamente, hasta la columna y fila $n-2$.

$A = \texttt{triGivens}(A)$
  $n = \texttt{dim}\,(A)$
  para $k = 1 : n - 2$
    para $i = n : -1 : k + 2$
      $[c, s] = \text{csGivens}(a_{i-1,k}\,,\, a_{ik})$
      $G = G(i - 1, i, c, s, n)$
      $A = G^{\mathrm{T}} A G$
    fin-para
  fin-para
fin `triHouse`

Ejemplo 9.15.

```
A  =
   2     3     4     5
   3    -1     0     1
   4     0    -2     8
   5     1     8     10

k = 1
i = 4
c = -0.624695    s = 0.780869

A  =
   2            3           -6.4031242    0
   3           -1           -0.7808688   -0.6246950
  -6.4031242   -0.7808688    13.121951    4.097561
   0           -0.6246950    4.097561    -5.1219512

i = 3
c = 0.424264    s = 0.905539

A  =
   2            7.0710678    0            0
   7.0710678    11.18       -4.9257204   -3.9755349
   0           -4.9257204    0.9419512    1.1727625
   0           -3.9755349    1.1727625   -5.1219512

k = 2
i = 4
```

```
c = 0.778168    s = -0.628057

A  =
   2           7.0710678    0           0
   7.0710678   11.18       -6.3298973   0
   0           -6.3298973   -0.3036510  -2.7160739
   0           0            -2.         -3.876349
```

No sobra recordar que el producto $G^{\mathrm{T}}AG$ debe ser hecho de manera eficiente, realizando únicamente las operaciones necesarias.

9.5.3 Valores propios de matrices tridiagonales simétricas

Sea T una matriz tridiagonal simétrica. En lo que sigue en esta sección, *se supone que T es tridiagonal simétrica.*

La matriz T puede ser el resultado del proceso de tridiagonalización de Householder o de Givens.

La matriz T se llama *no reducida*, [GoV96] pág. 416, si todos lo elementos subdiagonales (y los superdiagonales) son no nulos. Una matriz es *reducida* si algún elemento subdiagonal o (superdiagonal) es nulo.

Ejemplo 9.16. Una matriz no reducida y dos reducidas:

$$\begin{bmatrix} 2 & 3 & 0 & 0 & 0 & 0 \\ 3 & 4 & 5 & 0 & 0 & 0 \\ 0 & 5 & 6 & 7 & 0 & 0 \\ 0 & 0 & 7 & 8 & 9 & 0 \\ 0 & 0 & 0 & 9 & 10 & 11 \\ 0 & 0 & 0 & 0 & 11 & 12 \end{bmatrix}, \quad \begin{bmatrix} 2 & 3 & 0 & 0 & 0 & 0 \\ 3 & 4 & 5 & 0 & 0 & 0 \\ 0 & 5 & 6 & 7 & 0 & 0 \\ 0 & 0 & 7 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 11 \\ 0 & 0 & 0 & 0 & 11 & 12 \end{bmatrix}, \quad \begin{bmatrix} 2 & 3 & 0 & 0 & 0 & 0 \\ 3 & 4 & 5 & 0 & 0 & 0 \\ 0 & 5 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 11 \\ 0 & 0 & 0 & 0 & 11 & 12 \end{bmatrix}$$

T siempre se puede expresar como una matriz diagonal por bloques, donde cada bloque es de tamaño es 1×1 o de mayor tamaño pero tridiagonal no reducido.

En el primer caso del ejemplo anterior hay un solo bloque, en el segundo hay dos. En el tercer caso hay tres bloques, uno de ellos es 1×1.

Para encontrar los valores propios de T basta con encontrar los de cada bloque tridiagonal simétrico no reducido, agregando los bloques 1×1 que son valores propios de T.

El objetivo, a partir de ahora, es encontar los valores propios de T no reducida. Sea $T = QR$ la factorización QR de A y sea $T^+ = RQ$

$$\begin{aligned} Q^{\mathrm{T}}QR &= Q^{\mathrm{T}}T \\ R &= Q^{\mathrm{T}}T \\ T^+ = RQ &= Q^{\mathrm{T}}TQ. \end{aligned}$$

Luego T^+ es simétrica y semejante a T. Además se puede demostrar que también es tridiagonal.

Ejemplo 9.17.

```
T  =
   2     3     0     0
   3     4     5     0
   0     5     6     7
   0     0     7     8

R  =
  -3.6055513   -4.9923018   -4.1602515    0
   0           -5.0076864   -5.8371805   -6.9892556
   0            0           -7.6563459   -7.4712474
   0            0            0            2.8863072

Q  =
  -0.5547002   -0.0460830    0.3365427    0.7595545
  -0.8320503    0.0307220   -0.2243618   -0.5063697
   0           -0.9984651   -0.0224362   -0.0506370
   0            0           -0.9142743    0.4050957

T+  =
   6.1538462    4.1666469    0            0
   4.1666469    5.6743747    7.644594     0
   0            7.644594     7.0025484   -2.6388764
   0            0           -2.6388764    1.1692308
```

Un proceso, un poco lento, para hallar los valores propios de T, consiste en hacer $T = T^+$ y repetir varias veces. Se puede demostrar que la matriz que se va obteniendo tiende a ser reducida. Dicho en palabras populares, la tridiagonal se va adelgazando en alguna parte.

repetir
$QR = T$ factorización QR de T
$T = RQ$
fin-repetir

Ejemplo 9.18. Aplicar el proceso anterior hasta que T sea reducida. En este ejemplo se supone que T es reducida cuando para algún elemento subdiagonal $|t_{i+1,i}| \leq 10^{-10}$.

```
T  =
   2     3     0     0
   3     4     5     0
   0     5     6     7
   0     0     7     8

k = 1

T+  =
   9.8718663   -4.486006     0           0
  -4.486006     10.134151   -4.5625729   0
   0           -4.5625729   -1.1770851   -0.7764250
   0            0           -0.7764250    1.1710681


k = 2

T+  =
   13.296028   -3.5861468    0           0
  -3.5861468    8.2428763    1.7266634   0
   0            1.7266634   -2.7961816   0.3062809
   0            0            0.3062809   1.2572771


k = 10

T+  =
   15.191934   -0.0059687    0           0
  -0.0059687    6.6303783    0.0035727   0
   0            0.0035727   -3.100073    0.0002528
   0            0            0.0002528   1.2777606
```

```
 k = 20

 T+  =
    15.191938   -0.0000015    0            0
   -0.0000015    6.6303755    0.0000018    0
    0            0.0000018   -3.1000743    3.577E-08
    0            0            3.577E-08    1.2777606

 k = 27; matriz reducida:

T+  =
    15.191938   -4.514E-09    0            0
   -4.514E-09    6.6303755   -8.713E-09    0
    0           -8.713E-09   -3.1000743   -7.230E-11
    0            0           -7.229E-11    1.2777606
```

Denotemos por espec(A) el conjunto de valores propios de A. Cuando se hace un desplazamiento en los elementos diagonales de una matriz, los valores propios quedan desplazados igualmente, o sea,

$$\lambda \in \text{espec}(A) \quad \text{sssi} \quad \lambda - s \in \text{espec}(A - sI).$$

Hacer un desplazamiento adecuado en T puede acelerar notablemente la convergencia.

Ejemplo 9.19. Aplicar el mismo proceso a $T - sI$, con $s = 1$, hasta que para algún elemento $|t_{i+1,i}| \leq 10^{-10}$.

```
 T  =
    2     3     0     0
    3     4     5     0
    0     5     6     7
    0     0     7     8

 T - s I =
    1     3     0     0
    3     3     5     0
    0     5     5     7
    0     0     7     7
```

```
k = 9, matriz reducida:

T+  =
   14.191935   -0.0052796    0            0
  -0.0052796    5.5882374    0.6389663    0
   0            0.6389663   -4.057933    -8.844E-12
   0            0           -8.844E-12    0.2777606

T + s I
   15.191935   -0.0052796    0            0
  -0.0052796    6.5882374    0.6389663    0
   0            0.6389663   -3.057933    -8.844E-12
   0            0           -8.844E-12    1.2777606
```

Aunque hay varias maneras de calcular desplazamientos, uno de los más utilizados es el desplazamiento de Wilkinson

$$\begin{aligned} d &= t_{n-1,n-1} - t_{nn} \\ \mu &= t_{nn} + d - \text{signo}(d)\sqrt{d^2 + t_{n,n-1}^2} \\ &= t_{nn} - \frac{t_{n,n-1}^2}{d + \text{signo}(d)\sqrt{d^2 + t_{n,n-1}^2}} \end{aligned}$$

Para una matriz $T \in \mathbb{R}^{n\times n}$ tridiagonal, simétrica y no reducida, el proceso que se aplica es el siguiente:

mientras T sea no reducida
  cálculo de μ
  $T = T - \mu I$
  $QR = T$ factorización QR de T
  $T = RQ$
  $T = T + \mu I$
  para $i = 1 : n-1$
    si $|a_{i+1,i}| \leq \varepsilon\,(\,|a_{ii}| + |a_{i+1,i+1}|\,)$
      $a_{i+1,i} = 0$
      $a_{i,i+1} = 0$
    fin-si
  fin-para
fin-mientras

En [GoVa96], p. 420, se encuentra una descripción eficiente de la parte principal de este proceso, desde el cálculo de μ hasta $T = T + \mu I$.

Ejemplo 9.20. Hallar, por el proceso descrito anteriormente, una matriz tridiagonal semejante a la siguiente matriz tridiagonal:

$$T = \begin{bmatrix} 8 & 3 & 0 & 0 \\ 3 & 6 & -4 & 0 \\ 0 & -4 & -10 & -6 \\ 0 & 0 & -6 & 0 \end{bmatrix}$$

Con un propósito simplemente informativo, los valores propios obtenidos por la función `spec` son

$$-13.50417\,, \quad 1.9698954\,, \quad 5.0194039\,, \quad 10.51487$$

```
k = 1
 mu  =    2.8102497

 T  -mu I
    5.1897503    3            0           0
    3            3.1897503   -4           0
    0           -4           -12.81025   -6
    0            0           -6          -2.8102497

 T+ = RQ
    7.2885019    2.0988427    0            0
    2.0988427   -9.5701241    8.9042431    0
    0            8.9042431   -4.1976395   -0.6390185
    0            0           -0.6390185   -0.7617370

 T + mu I
    10.098752    2.0988427    0            0
    2.0988427   -6.7598744    8.9042431    0
    0            8.9042431   -1.3873898   -0.6390185
    0            0           -0.6390185    2.0485127


 k = 2
 mu  =    2.1635102
```

```
T  -mu I
   7.9352413    2.0988427    0            0
   2.0988427   -8.9233846    8.9042431    0
   0            8.9042431   -3.5509      -0.6390185
   0            0           -0.6390185   -0.1149975

T+ = RQ
   7.8706324   -3.26714      0            0
  -3.26714     -14.885642   -2.4468061    0
   0           -2.4468061    2.5541744    0.0357613
   0            0            0.0357613   -0.1932052

T + mu I
   10.034143   -3.26714      0            0
  -3.26714     -12.722132   -2.4468061    0
   0           -2.4468061    4.7176845    0.0357613
   0            0            0.0357613    1.970305


k = 3
mu  =  1.9698396

T  -mu I
   8.064303   -3.26714      0            0
  -3.26714    -14.691972   -2.4468061    0
   0          -2.4468061    2.7478449    0.0357613
   0           0            0.0357613    0.0004654

T+ = RQ
   7.1298463    5.6488809    0            0
   5.6488809   -14.048752    0.5009906    0
   0            0.5009906    3.0394919    0.0000006
   0            0            0.0000006    0.0000557

T + mu I
   9.0996859    5.6488809    0            0
   5.6488809   -12.078913    0.5009906    0
   0            0.5009906    5.0093315    0.0000006
   0            0            0.0000006    1.9698954
```

```
k = 4
mu  = 1.9698954

T  -mu I
   7.1297905    5.6488809    0            0
   5.6488809   -14.048808    0.5009906    0
   0            0.5009906    3.0394362    0.0000006
   0            0            0.0000006    1.379E-13

T+ = RQ
   4.4614948   -9.0220625    0            0
  -9.0220625   -11.390431   -0.1052167    0
   0           -0.1052167    3.049355    -2.585E-17
   0            0            1.656E-22    7.811E-16

T + mu I
   6.4313901   -9.0220625    0            0
  -9.0220625   -9.4205358   -0.1052167    0
   0           -0.1052167    5.0192503   -2.585E-17
   0            0            1.656E-22    1.9698954

T reducida
   6.4313901   -9.0220625    0            0
  -9.0220625   -9.4205358   -0.1052167    0
   0            0.1052167    5.0192503    0
   0            0            0            1.9698954
```

En una matriz simétrica tridiagonal se busca desde la esquina S.E. hacia la esquina N.O., el primer bloque de tamaño superior a uno que sea no reducido. A este bloque se le aplica el procedimiento anterior (hasta que el bloque sea reducido). El proceso general acaba cuando la matriz resultante es diagonal.

Ejemplo 9.21. Obtener los valores propios de la siguiente matriz tridiagonal simétrica:

$$A = \begin{bmatrix} -2 & 8 & 0 & 0 & 0 & 0 \\ 8 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 3 & 0 & 0 \\ 0 & 0 & 3 & 6 & -4 & 0 \\ 0 & 0 & 0 & -4 & -10 & -6 \\ 0 & 0 & 0 & 0 & -6 & 0 \end{bmatrix}$$

```
i1 i2 :   3   6

 T inicial
    8     3     0      0
    3     6    -4      0
    0    -4    -10    -6
    0     0    -6      0
 mu = 2.810250

 T final
    10.098752    2.0988427    0            0
    2.0988427   -6.7598744    8.9042431    0
    0            8.9042431   -1.3873898   -0.6390185
    0            0           -0.6390185    2.0485127

i1 i2 :   3   6

 T inicial
    10.098752    2.0988427    0            0
    2.0988427   -6.7598744    8.9042431    0
    0            8.9042431   -1.3873898   -0.6390185
    0            0           -0.6390185    2.0485127
 mu = 2.163510

 T final
    10.034143   -3.26714      0            0
   -3.26714     -12.722132   -2.4468061    0
    0           -2.4468061    4.7176845    0.0357613
    0            0            0.0357613    1.970305

i1 i2 :   3   6
```

```
 T inicial
    10.034143   -3.26714      0            0
   -3.26714     -12.722132   -2.4468061    0
    0           -2.4468061    4.7176845    0.0357613
    0            0            0.0357613    1.970305
 mu = 1.969840

 T final
    9.0996859    5.6488809    0            0
    5.6488809   -12.078913    0.5009906    0
    0            0.5009906    5.0093315    0.0000006
    0            0            0.0000006    1.9698954

i1 i2 :   3   6

 T inicial
    9.0996859    5.6488809    0            0
    5.6488809   -12.078913    0.5009906    0
    0            0.5009906    5.0093315    0.0000006
    0            0            0.0000006    1.9698954
 mu = 1.969895

 T final
    6.4313901   -9.0220625    0            0
   -9.0220625   -9.4205358   -0.1052167    0
    0           -0.1052167    5.0192503    8.383E-17
    0            0           -1.058E-22    1.9698954

 A  =
   -2     8     0            0            0            0
    8    -2     0            0            0            0
    0     0     6.4313901   -9.0220625    0            0
    0     0    -9.0220625   -9.4205358   -0.1052167    0
    0     0     0           -0.1052167    5.0192503    0
    0     0     0            0            0            1.9698954

i1 i2 :   3   5

 T inicial
    6.4313901   -9.0220625    0
```

```
   -9.0220625   -9.4205358   -0.1052167
    0           -0.1052167    5.0192503
 mu = 5.020017

 T final
   -6.2865541    11.012094    0
    11.012094    3.2972548   -0.0000058
    0           -0.0000058    5.0194039

i1 i2 :   3   5

 T inicial
   -6.2865541    11.012094    0
    11.012094    3.2972548   -0.0000058
    0           -0.0000058    5.0194039
 mu = 5.019404

 T final
   -12.629095   -4.5002992    0
   -4.5002992    9.6397959    2.575E-17
    0            2.079E-17    5.0194039

 A  =
   -2     8     0            0            0            0
    8    -2     0            0            0            0
    0     0    -12.629095   -4.5002992    0            0
    0     0    -4.5002992    9.6397959    0            0
    0     0     0            0           5.0194039    0
    0     0     0            0           0            1.9698954

i1 i2 :   3   4

 T inicial
   -12.629095   -4.5002992
   -4.5002992    9.6397959
 mu = 10.514870

 T final
   -13.50417    -2.914E-16
    3.384E-16    10.51487
```

```
 A  =
   -2     8     0           0           0           0
    8    -2     0           0           0           0
    0     0    -13.50417    0           0           0
    0     0     0           10.51487    0           0
    0     0     0           0           5.0194039   0
    0     0     0           0           0           1.9698954

i1 i2 :   1   2

 T inicial
   -2     8
    8    -2
 mu = -10.000000

 T final
    6           -8.782E-17
   -1.735E-18   -10

 A  =
    6     0     0           0           0           0
    0   -10     0           0           0           0
    0     0    -13.50417    0           0           0
    0     0     0           10.51487    0           0
    0     0     0           0           5.0194039   0
    0     0     0           0           0           1.9698954
```

En los resultados anteriores, `i1` e `i2` indican la fila inicial y final de la primera submatriz no reducida que se encuentra y con la que se va a trabajar.

Bibliografía

[AlK02] Allaire G. y Kaber S.M., *Algèbre linéaire numérique*, Ellipses, Paris, 2002.

[Atk78] Atkinson Kendall E., *An Introduction to Numerical Analysis*, Wiley, New York, 1978.

[BuF85] Burden R.L. y Faires J.D., *Numerical Analysis*, 3a. ed., Prindle-Weber-Schmidt, Boston, 1985.

[Dem97] Demmel J.W., *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[GoV96] Golub G.H. y Van Loan C.H., *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.

[KiC94] Kincaid D. y Cheney W., *Análisis numérico*, Addison-Wesley Iberoamericana, Wilmington, 1994.

[Man04] Mantilla I., *Análisis Numérico*, Universidad Nacional, Fac. de Ciencias, Bogotá, 2004

[Par80] Parlett B.N. *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, 1980.

[Ste98] Stewart G.W., *Matrix Algorithms, Volume I: Basic Decompositions*, Siam, Philadelphia, 1998.

ÍNDICE ANALÍTICO