MÉTODOS NUMÉRICOS EN INGENIERÍA

Luis Alvarez León

Curso 2023 - 2024

Email: lalvarez.mat@gmail.com
WWW: https://sites.google.com/site/luisalvarezsite/

Prólogo

En este libro se desarrollan los contenidos de una asignatura básica de Métodos Numéricos orientada a estudios universitarios de ingeniería o cualquier otra disciplina donde se conceda importancia al diseño y programación de algoritmos numéricos. En cada uno de los temas, que se describen a continuación, se hace especial énfasis en la parte algorítmica y de programación, destacando para cada método su algoritmo asociado. Además de una colección de problemas resueltos para cada tema, se añade un acceso, por internet, a ejercicios de programación en C++ sobre los diferentes algoritmos. Se han incluido diversas aplicaciones de los métodos numéricos en Epidemilogía, tratamiento de imágenes y gráficos por ordenador.

En el primer tema se tratan las aritméticas de precisión finita. Se construye manualmente una aritmética muy sencilla para ilustrar los conceptos de mantisa, exponente y las operaciones básicas dentro de la aritmética. Además se hace especial énfasis en que algunas propiedades algebraicas básicas de los números reales no se cumplen al trabajar en aritméticas de precisión finita. Se estudian las aritméticas estándar, los errores de redondeo y la manera correcta de comparar variables dentro de los algoritmos.

En el segundo tema se presentan los métodos habituales para el cálculo de ceros de funciones de una variable, como son los métodos de la bisección, regula-falsi, Newton-Raphson, secante y Müller. Para cada método se presenta de forma clara el algoritmo asociado incluyendo sus posibles criterios de parada. Se estudia también el caso particular de los polinomios, mostrando un algoritmo básico que permite calcular todas las raíces reales de un polinomio.

En el tercer tema se estudia la interpolación de funciones, en primer lugar se presenta la interpolación por intervalos, estudiando los casos de los splines de grados 0, 1, 2 y 3. En segundo lugar se estudia la interpolación de Lagrange, las diferencias de Newton para calcular el polinomio interpolador, el error de interpolación y la aproximación de funciones elementales usando polinomios. A continuación, por su importancia, se introduce brevemente la regresión lineal interpretada como una técnica de interpolación. Finalmente se estudia la interpolación en 2D con una aplicación al procesado de imágenes.

En el tema 4 se estudian los métodos directos básicos para la resolución de sistemas de ecuaciones lineales como son: los métodos de descenso/remonte para las matrices triangulares, el método de Gauss, las factorizaciones de Cholesky y LU y el método de Crout para matrices tridiagonales. También se estudian métodos para calcular el determinante de una matriz y su inversa. Se estudia con cierto detalle la complejidad computacional de los diferentes métodos.

La primera parte del tema 5 se dedica a la diferenciación numérica. Se presentan las fórmulas estándar para la aproximación de la primera y segunda derivada y su fórmula de error. Dado un punto x y una función $f(x)$, se estudia los posibles valores del paso h que se pueden tomar al evaluar $f(x+h)$ en función de la magnitud de x y de la precisión de la aritmética. Se estudia el cálculo numérico de derivadas en dimensión 2 usando máscaras de convolución 3×3. Se presenta una aplicación al procesado de imágenes digitales y en particular al cálculo de bordes. En la segunda parte de este tema se presentan las técnicas básicas de integración numérica incluyendo la estrategia de dividir el intervalo de integración en subintervalos y aplicar las técnicas de aproximación del rectángulo, trapecio o Simpson en cada subintervalo. Así como las técnicas de cuadratura que se basan en buscar la exactitud de la fórmula de integración para polinomios. Se estudia también la integración en $2D$ en recintos rectangulares.

En el tema 6 se tratan cuestiones más avanzadas del Análisis Matricial. Se estudia el condicionamiento de una matriz como el número que determina la bondad de una matriz para resolver sistemas a partir de ella, la técnica de Jacobi para el cálculo de los autovalores y autovectores de matrices simétricas y los métodos de la potencia para calcular el autovalor máximo, mínimo y el más cercano a un número dado. A continuación se presentan los métodos iterativos habituales, es decir, los métodos de Jacobi, Gauss-Seidel y relajación que están especialmente adaptados a la resolución de sistemas con matrices escasas que poseen un gran número de ceros. Finalmente se estudia el método de Newton-Raphson para resolver sistemas no-lineales de cualquier dimensión. A continuación se introduce la optimización como la minimización de una función objetivo usando como ejemplo la regresión lineal, se estudia la aplicación del Método de Newton Raphson a la optimización y se introducen los métodos de gradiente descendente atenuado y el método de Newton-Raphson atenuado.

En el tema 7 se tratan cuestiones más avanzadas de la interpolación, como la interpolación de Hermite, que incluye la interpolación de los valores de las derivadas, la interpolación por la función seno cardinal y por polinomios trigonométricos de gran importancia en la teoría de Fourier y en aplicaciones como el sonido digital.

En el apéndice se dan las indicaciones necesarias para acceder y trabajar con los ejercicios de programación en C++ que se proponen. Las aplicaciones en Epidemiología que se presentan también están implementadas en estos ejercicios.

Por último se presenta la bibliografía básica utilizada en la confección de este libro.

Índice general

Capítulo 1

ARITMÉTICAS DE PRECISIÓN FINITA

1.1. Introducción

Los números se gestionan en el ordenador usando aritméticas de precisión finita donde se dedica un número finito de bits para almacenar tanto los números enteros como los reales. La calidad de una aritmética de precisión finita viene dada por el número de bits que usa para el almacenamiento y la calidad/velocidad con las que realiza operaciones internas como sumas, productos, etc.. Esta calidad puede tener una trascendencia enorme en los resultados de los algoritmos numéricos donde se realizan una gran cantidad de operaciones con números. Por ejemplo, el 4 de junio de 1996, el cohete Ariane 5 se desvió de su trayectoria de despegue y explotó a causa de errores en los cálculos debidos a que se usó una aritmética de precisión finita diseñada para un modelo anterior de cohete (el Ariane 4) que no era adecuada para las características del nuevo cohete que por ser más rápido y potente requería más precisión y velocidad en los cálculos del sistema de guiado del cohete.

A continuación estudiaremos como se representan los números en el ordenador. Un número entero k se representa en el ordenador en base 2 a través de un número fijo de bits (habitualmente 16 o 32 bits), donde uno de los bits se utiliza para determinar el signo y los restantes para expresar la magnitud del número. Es decir un número entero se representa como

$$k = \pm \left(a_1 + a_2 2 + a_3 2^2 + ... + a_n 2^{n-1}\right), \tag{1.1}$$

donde $a_i = 0$ o $a_i = 1$, $n = 15$ en el caso de 16 bits y $n = 31$ en el caso de 32 bits. De esta forma, el mayor número entero que se puede alcanzar en la aritmética es

$$\begin{aligned} &16 \text{ bits } \rightarrow k_{max} = 1 + 2 + 2^2 + ... + 2^{14} = 2^{15} - 1 = 32{,}767 \\ &32 \text{ bits } \rightarrow k_{max} = 1 + 2 + 2^2 + ... + 2^{30} = 2^{31} - 1 = 2{,}147{,}483{,}647. \end{aligned}$$

Para introducir como se representan los números reales en el ordenador vamos primero a analizar como se representan en base 10 que es la forma habitual en que

los manejamos mentalmente. Por ejemplo, usando la notación científica, el número 654.3 puede escribirse como

$$654.3 = 10^2 \cdot 6.543 = 10^2 \left(6 + \frac{5}{10} + \frac{4}{10^2} + \frac{3}{10^3}\right), \tag{1.2}$$

de forma general, un número real z distinto de cero puede representarse como

$$z = \pm 10^e \sum_{n=0}^{\infty} \frac{a_n}{10^n} = \pm 10^e \left(a_0 + \frac{a_1}{10} + \frac{a_2}{10^2} + \ \cdots \ + \ \cdots\right), \tag{1.3}$$

donde e es un número entero que representa el exponente y la secuencia $a_0, a_1, ...$ son números naturales que toman valores entre 0 y 9 ($a_0 > 0$) y que representan la mantisa. Para almacenar los números en el ordenador en forma binaria es más adecuado que los números a_n se muevan entre 0 y 1. Para ello se usa una representación en base 2 donde los números reales distintos de cero se almacenan como

$$y = \pm 2^e \left(1 + \sum_{n=1}^{\infty} \frac{a_n}{2^n}\right) = \pm 2^e \left(1 + \frac{a_1}{2} + \frac{a_2}{2^2} + \ \cdots \ + \ \cdots\right), \tag{1.4}$$

donde e es el exponente que es un número entero y la secuencia $a_1 a_2 a_3 ...$ con $a_n \in \{0, 1\}$ se denomina mantisa.

Ejemplo 1 *Consideremos* $z = 10.5$, *vamos a representarlo en base 2. En primer lugar calculamos el exponente que viene dado por el mayor número entero tal que* $2^e \leq z$. *En nuestro caso sale* $e = 3$. *A continuación calculamos*

$$\frac{10.5}{2^3} = 1.3125. \tag{1.5}$$

Para calcular a_n *vamos progresivamente asignándole inicialmente el valor* $a_n = 1$, *calculamos el resultado y si es mayor que* 1.3125 *entonces tomamos* $a_n = 0$, *es decir, vamos haciendo*

$$\begin{aligned}
1 + \frac{1}{2} &= 1.5 > 1.3125 \rightarrow a_1 = 0, \\
1 + \frac{1}{2^2} &= 1.25 < 1.3125 \rightarrow a_2 = 1, \\
1 + \frac{1}{2^2} + \frac{1}{2^3} &= 1.375 > 1.3125 \rightarrow a_3 = 0, \\
1 + \frac{1}{2^2} + \frac{1}{2^4} &= 1.3125 = 1.3125 \rightarrow a_4 = 1,
\end{aligned}$$

como ya hemos encontrado el valor exacto del número obtenemos que se puede expresar como

$$10.5 = 2^3(1 + \frac{0}{2} + \frac{1}{2^2} + \frac{0}{2^3} + \frac{1}{2^4} + \frac{0}{2^5} + .. + \frac{0}{2^n} + ..), \tag{1.6}$$

a efectos de almacenamiento del número en el ordenador, como el primer elemento de la mantisa es siempre igual a 1 no se almacena y por tanto la mantisa de este número vendría dada por la secuencia binaria

$$010100000... \tag{1.7}$$

Nótese que en el ejemplo anterior, el número de elementos a_n distintos de 0 es finito, en general no siempre es así. Por ejemplo, si un número es irracional siempre tiene un número infinito de elementos a_n no nulos. Por otro lado, un número que en base 10 tiene un número finito de decimales al expresarlo en base 2 puede tener un número infinito de elementos a_n no nulos como muestra el siguiente resultado

Teorema 1 *Al representar el número real* 0.1 *como*

$$0.1 = 2^e \sum_{n=0}^{\infty} \frac{a_n}{2^n}, \tag{1.8}$$

el número de elementos no nulos a_n es infinito.

Demostración: Supongamos que para algún t finito y e entero se tiene:

$$0.1 = 2^e \sum_{n=0}^{t} \frac{a_n}{2^n}, \tag{1.9}$$

despejando en esta igualdad obtenemos

$$2^{t-e} = 10 \sum_{n=0}^{t} a_n 2^{t-n}, \tag{1.10}$$

ahora bien, como el número $m = \sum_{n=0}^{t} a_n 2^{t-n}$ es entero, de la igualdad anterior obtenemos

$$2^{t-e} = 5 \cdot 2m, \tag{1.11}$$

pero esta igualdad implica que el número 2^{t-e} es divisible por 5 lo cual es imposible.

Para definir una aritmética de precisión finita de número reales, lo que se hace es asignar un rango finito de valores al exponente e y un número finito $a_1, ..., a_t$ de coeficientes para la mantisa. Por tanto, en una aritmética de precisión finita, los números reales distintos de cero se representan como

$$\widetilde{z} = \pm 2^e \left(1 + \sum_{n=1}^{t} \frac{a_n}{2^n} \right), \tag{1.12}$$

donde el exponente e es un número entero que varía entre dos valores limites $e_{min} \leq e \leq e_{max}$ y que determina la magnitud del número. Al valor t se le llama precisión

de la aritmética. La mantisa, dada por la secuencia $a_1a_2.....a_t$, (donde $a_n \in \{0,1\}$), determina la precisión con la se almacenan los números. Es importante observar que el número 0 debemos añadirlo a la aritmética ya que 0 no se puede representar de la forma anterior (debido a la existencia siempre del número 1 dentro del paréntesis). La calidad de una aritmética depende del número de bits que se usen para almacenar el número y de como se reparten esos bits para almacenar el exponente y la mantisa. Para distinguir los números que están en la aritmética de los que no están usaremos la notación $\widetilde{z}$ para indicar los números que pertenecen a la aritmética, es decir que se pueden expresar de forma exacta con la fórmula anterior. Dado un número real cualquiera z, $\widetilde{z}$ representa el número de la aritmética que está más cerca de z.

Definición 1 *Dada una aritmética de precisión finita, se define la unidad de redondeo u como*

$$u = 2^{-t}, \tag{1.13}$$

donde t es la precisión de la aritmética, es decir, el número de bits dedicado a almacenar la mantisa.

Como veremos posteriormente, la unidad de redondeo, u, es determinante para calcular el error de redondeo que viene dado por la diferencia entre un número real z y su aproximación en la aritmética $\tilde{z}$.

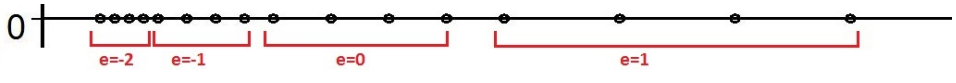
Para hacernos una idea de como funciona una aritmética de precisión finita vamos a calcular todos los números positivos de una aritmética de 5 bits construida de la forma siguiente: se usa 1 bit para el signo, 2 bits para la mantisa (es decir $t = 2$) y 2 bits para el exponente e, tomando como rango de exponentes $e = -1, 0, 1, 2$. Es decir los números positivos que se pueden generar tienen la forma

$$\widetilde{z} = 2^e \left(1 + \frac{a_1}{2} + \frac{a_2}{2^2}\right), \tag{1.14}$$

en la siguiente tabla se muestran todos los posibles números positivos generados con esta aritmética en función de los diferentes valores posibles del exponente e.

e	$a_1 = a_2 = 0$	$a_1 = 0, a_2 = 1$	$a_1 = 1, a_2 = 0$	$a_1 = a_2 = 1$
-2	$\frac{1}{2^2} = 0.25$	$\frac{1}{2^2} + \frac{1}{2^4} = 0.3125$	$\frac{1}{2^2} + \frac{1}{2^3} = 0.375$	$\frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} = 0.4375$
-1	$\frac{1}{2} = 0.5$	$\frac{1}{2} + \frac{1}{2^3} = 0.625$	$\frac{1}{2} + \frac{1}{2^2} = 0.75$	$\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} = 0.875$
0	1	$1 + \frac{1}{2^2} = 1.25$	$1 + \frac{1}{2} = 1.5$	$1 + \frac{1}{2} + \frac{1}{2^2} = 1.75$
1	2	$2 + \frac{1}{2} = 2.5$	$2 + 1 = 3$	$2 + 1 + \frac{1}{2} = 3.5$

Si representamos estos números sobre una recta obtenemos



Es importante resaltar que los números reales en una aritmética de precisión finita no son equidistantes, observamos que en cada intervalo de la forma $[2^e, 2^{e+1})$

hay $2^t = 4$ números que están a una distancia entre sí dada por $\frac{2^e}{2^t} = \frac{2^e}{4}$. Por tanto los números están más cercanos entre sí cerca de 0, y más alejados al separarnos de 0. El total de números de la aritmética viene dado por los números de la tabla anterior, los mismos números con signo negativo y el número 0, que hay que añadir como un caso especial al no estar en la tabla.

Al realizar una operación en una aritmética de precisión finita (sumas, multiplicaciones, etc.), si se almacena el resultado en la propia aritmética se redondea al valor más cercano existente en la aritmética. En la siguiente tabla se muestran algunos resultados de operaciones en la aritmética de ejemplo anterior después de realizar el redondeo. En el caso en que el resultado esté a la misma distancia de 2 números de la aritmética se toma el menor de ellos (es decir se redondea por debajo).

$2 + 0.25 = 2$	$(2 + 0.25) + 0.25 = 2$	$2 + (0.25 + 0.25) = 2.5$
$0.5 \cdot 0.25 = 0$	$2 \cdot (0.5 \cdot 0.25) = 0$	$(2 \cdot 0.5) \cdot 0.25 = 0.25$
$\frac{1}{1.75} = 0.5$	$1.75 \left(\frac{1}{1.75}\right) = 0.875$	$2 \cdot 3.5 = 3.5$

Como puede verse en estos ejemplos existen diferencias importantes entre como se comportan las operaciones con números reales en precisión infinita (todos los números reales) y en aritméticas de precisión finita. En la siguiente tabla se muestran algunas de estas diferencias

Precisión infinita	Precisión finita
$x + y = x \Longrightarrow y = 0$	$x + y = x \nRightarrow y = 0$
$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	$x \cdot (y \cdot z) \neq (x \cdot y) \cdot z$
$x + (y + z) = (x + y) + z$	$x + (y + z) \neq (x + y) + z$
$x \neq 0 \Longrightarrow x \cdot \left(\frac{1}{x}\right) = 1$	$x \neq 0 \nRightarrow x \cdot \left(\frac{1}{x}\right) = 1$
x positivo $\Longrightarrow 2x > x$	x positivo $\nRightarrow 2x > x$
$x > 0 \Longrightarrow \frac{x}{2} > 0$	$x > 0 \nRightarrow \frac{x}{2} > 0$

1.2. Las aritméticas estándar

En 1985, la sociedad I.E.E.E. presentó una serie de especificaciones estándares para la definición de una aritmética de precisión finita para los números reales. En este trabajo, se codifica un número real en simple precisión utilizando 32 bits de memoria, de los cuales 23 bits se utilizan para la mantisa (es decir $t = 23$), 1 bit se utiliza para el signo y 8 bits se utilizan para el exponente e, lo cual da un rango de $2^8 = 256$ valores posibles para el exponente e. En este caso, se toma $e_{min} = -125$ y $e_{max} = 128$. Como puede observarse, el número total de exponentes posibles es 254, dos menos que los 256 posibles, ello se hace así porque se reservan dos casos para tratar las denominadas excepciones, como se verá más adelante. También se define en este trabajo de I.E.E.E. un estándar para una aritmética en doble precisión. En este caso, se utilizan 64 bits para almacenar un número real, de los cuales 52 bits se

utilizan para la mantisa ($t = 52$), 1 bit para el signo y 11 bits para el exponente, lo que da lugar a $2^{11} = 2048$ posibilidades de elección de exponente e. En este caso, se toma $e_{min} = -1021$ y $e_{max} = 1024$. Además de estas aritméticas de 32 y 64 bits existen, aunque se usan menos, otras aritméticas con mayor número de bits. Por ejemplo, el lenguaje C/C++ nos ofrece los siguiente tipos de variables reales.

TIPO	bits	bits t	bits e	u=$1/2^t$	n^o menor	n^o mayor
float	32	23	8	$\approx 1.19 \times 10^{-7}$	$\approx 10^{-45}$	$\approx 10^{38}$
double	64	52	11	$\approx 2.22 \times 10^{-16}$	$\approx 10^{-324}$	$\approx 10^{308}$
long double	80	63	16	$\approx 1.08 \times 10^{-19}$	$\approx 10^{-4937}$	$\approx 10^{4928}$

Un resultado importante es que en cada intervalo $[2^e, 2^{e+1})$ los números de la aritmética se reparten uniformemente estando a una distancia $2^e u$ entre ellos. Es decir, estos números se pueden expresar como

$$2^e + k \cdot 2^e u = 2^e(1 + k \cdot u) \qquad k = 0, 1, 2, .., 2^t - 1, \tag{1.15}$$

por ejemplo, en la aritmética de 32 bits, en el intervalo $[1, 2)$ los puntos están a una distancia entre sí de apróximadamente 1.19×10^{-7} y en 64 bits a una distancia de 2.22×10^{-16}. Si, por ejemplo ejecutamos la instrucción $A = 1. + \frac{1}{2^{24}}$ en precisión float en la variable A se guardará el número 1. Si ejecutamos $A = 1.+0.1$ el resultado será mayor que 1 pero habrá un redondeo debido a que 0.1 no se puede representar de forma exacta.

1.3. Tratamiento de las excepciones en el estándar de I.E.E.E.

En una aritmética de precisión finita existen 3 excepciones que debemos tratar de forma especial: el número 0, el infinito y operaciones no válidas, NaN, (como $\sqrt{-1}$). La diferencia fundamental entre la excepción infinito y NaN es que con la excepción infinito todavía se pueden seguir haciendo operaciones. Por ejemplo si hacemos las operaciones A=1/0 y a continuación B=1/A, la variable B será igual a cero. Sin embargo esta operación no tiene sentido si A=$\sqrt{-1}$. Las excepciones son tratadas en el estándar de I.E.E.E. de la siguiente forma: dentro de las posiciones de memoria dedicadas al exponente e de un número, se reservan dos, que corresponden a $e_{min} - 1$ y $e_{max} + 1$, para trabajar con las excepciones. La regla que se utiliza es la siguiente:

1. Si el valor de una variable $\tilde{z}$ tiene por exponente $e_{max}+1$ y todos los coeficientes de la mantisa valen 0, entonces $\tilde{z}$ se considera infinito. Por ejemplo 1/0 debe dar infinito.

2. Si el valor de una variable $\tilde{z}$ tiene por exponente $e_{max} + 1$ y algún coeficiente de la mantisa es distinto de 0, entonces y se considera que no es un número (NaN (Not a Number)). Por ejemplo $\sqrt{-1}$ debe dar NaN.

3. Si el valor de una variable $\tilde{z}$ tiene por exponente $e_{min}-1$ y todos los coeficientes de la mantisa valen 0, entonces $\tilde{z}$ se considera igual a 0.

4. Si el valor de una variable $\tilde{z}$ tiene por exponente $e_{min}-1$ y algún coeficiente de la mantisa es distinto de 0, $\tilde{z}$ se considera que no está normalizado (es decir $a_0 = 0$) y el valor de $\tilde{z}$ sería

$$\tilde{z} = 2^{e_{min}-1} \sum_{n=1}^{t} \frac{a_n}{2^n}, \tag{1.16}$$

por tanto, si se tienen en cuenta las excepciones, el número positivo más pequeño de la aritmética sería

$$\tilde{z} = 2^{e_{min}-1} \frac{1}{2^t}, \tag{1.17}$$

en la práctica no todas las aritméticas tienen implementada esta opción. Por ello, salvo que se indique que se están usando excepciones consideramos que el número más pequeño es $2^{e_{min}}$.

1.4. Fuentes de errores numéricos.

Error de cambio de base

Este tipo de error es un tipo especial de error de redondeo que se produce al realizar un cambio de base para representar un número real. Como vimos en la sección anterior, las aritméticas estándar de ordenador trabajan en base 2. Sin embargo, los humanos pensamos y razonamos en términos de números en base 10. Por ejemplo, números tan usuales para nosotros como 0.1 no pueden representarse de forma exacta en una aritmética en base 2. Esto quiere decir que, al representar 0.1 en el ordenador, se va a producir un pequeño redondeo, y este pequeño error de redondeo se puede ir propagando hasta producir errores apreciables. Por ejemplo, parece razonable pensar que, cuando sumamos $10,000$ veces el número 0.0001, el resultado sea exactamente 1, pero, no es así. Sin embargo, si sumamos $16,384 = 2^{14}$ veces el número 2^{-14}, el resultado si es exactamente 1. A nivel algorítmico, estos errores de redondeo pueden provocar que la condición de parada en un bucle falle. Por ejemplo, si en un bucle tomamos una variable $\tilde{z} = 0$, dentro del bucle hacemos la operación $\tilde{z} = \tilde{z} + 0.0001$ y ponemos como condición de parada del bucle que $\tilde{z}$ sea igual a 1, puede ser que la condición de parada nunca se cumpla.

Como conclusión de este apartado, podemos extraer que, para ser más precisos numéricamente, cuando trabajamos con números más pequeños que la unidad deberíamos pensar en términos de 2^{-m} en lugar de 10^{-m}, que es como solemos hacerlo. De hecho, en algunos de los apartados de este tema se ponen ejemplos usando números en base 10 cuando sería más apropiado hacerlo en base 2; pero se ponen en base 10 para facilitar su comprensión.

Error de cancelación

Estos errores se producen al restar números de aproximadamente la misma magnitud. Hay que tener en cuenta que, al realizar operaciones sobre una variable, los errores de redondeo se van acumulando en la parte menos significativa del número (los últimos elementos de la mantisa), dejando relativamente intacta la parte más significativa del número, que corresponde a los primeros elementos de la mantisa. Por ello, al restar dos números de magnitud parecida, se cancelan las partes significativas, quedando la aportación de los dígitos de menos valor, que es donde más error hay. Al diseñar algoritmos, en la medida de lo posible, se debe evitar la posibilidad de restar 2 números que pudieran ser de magnitud parecida. Por ejemplo, en la conocida fórmula del cálculo de raíces de un polinomio de grado 2, $ax^2+bx+c=0$ (con $a \neq 0$)

$$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}, \tag{1.18}$$

una forma de evitar la cancelación que se produce cuando $b \approx \sqrt{b^2-4ac}$ consiste en calcular primero la raíz de mayor valor absoluto, es decir

$$x_1 = \frac{-\left(b + sign(b)\sqrt{b^2-4ac}\right)}{2a}, \tag{1.19}$$

y después la segunda raíz x_2 utilizando la relación $x_1 x_2 = \frac{c}{a}$. $sign(b)$ es la función signo que se define como $sign(b) = 1$ si $b > 0$ y $sign(b) = -1$ si $b < 0$.

Error de redondeo

El error de redondeo se produce, principalmente, cuando intentamos almacenar en una variable de la aritmética un número cuya mantisa tiene un número de bits que supera el número de bits máximo (que hemos llamado t) permitido de la aritmética. En ese caso, el número se sustituye por el más cercano dentro de la aritmética. En general, las operaciones básicas como sumas, restas, multiplicaciones y divisiones producen errores de redondeo al almacenar los resultados. Vamos a llamar A al conjunto de valores reales a los que da lugar una aritmética de precisión finita, es decir

$$A = \left\{ \pm 2^e \left(1 + \sum_{n=1}^{t} \frac{a_n}{2^n} \right) \right\} \cup \{0\}. \tag{1.20}$$

Dado un número real cualquiera z, llamaremos $\widetilde{z} \in A$ al número de la aritmética más cercano a z dentro de A. A continuación, mostraremos un resultado que indica el error de redondeo máximo que se produce al aproximar un número real cualquiera en una aritmética de precisión finita.

Teorema 2 *Sea $u = 2^{-t}$ la unidad de redondeo y $e \in Z$ tal que $e_{min} \leq e \leq e_{max}$. Si un número real z verifica que $\mid z \mid \in [2^e, 2^{e+1})$, entonces*

$$\mid z - \widetilde{z} \mid \leq \mid z \mid \frac{u}{2} \tag{1.21}$$

donde $\widetilde{z}$ es el número más cercano a z en la aritmética.

Demostración. Supondremos, sin perdida de generalidad, que $z > 0$. Por lo visto anteriormente los puntos de la aritmética se distribuyen uniformemente en $[2^e, 2^{e+1})$ a una distancia entre ellos de $2^e u$. Como $z \in [2^e, 2^{e+1})$ entonces $\tilde{z} \in [2^e, 2^{e+1}]$ y por tanto debe cumplir:

$$| z - \widetilde{z} | \leq \frac{2^e u}{2} \leq | z | \frac{u}{2}, \tag{1.22}$$

con lo que queda demostrado el teorema. A continuación presentamos un resultado que indica que si la distancia entre dos puntos de la aritmética es muy pequeña, entonces los puntos tienen que coincidir.

Teorema 3 *Sean $\widetilde{z}_1$, $\widetilde{z}_2$ dos números de una aritmética de precisión finita con $\widetilde{z}_1 \neq 0$ entonces:*

$$| \widetilde{z}_1 - \widetilde{z}_2 | < | \widetilde{z}_1 | \frac{u}{2} \quad \Rightarrow \quad \widetilde{z}_1 = \widetilde{z}_2, \tag{1.23}$$

siendo u la unidad de redondeo.

Demostración. Consideramos sin perdida de generalidad que $\widetilde{z}_1 > 0$. Sea e el exponente tal que $\widetilde{z}_1 \in [2^e, 2^{e+1})$. Por lo visto anteriormente los puntos de la aritmética se distribuyen uniformemente en $[2^e, 2^{e+1})$ a una distancia entre ellos de $2^e u$. Por tanto si $\widetilde{z}_2 \in [2^e, 2^{e+1})$

$$| \widetilde{z}_1 - \widetilde{z}_2 | < 2^e u \quad \Rightarrow \quad \widetilde{z}_1 = \widetilde{z}_2, \tag{1.24}$$

por otro lado

$$| \widetilde{z}_1 | \frac{u}{2} < 2^{e+1} \frac{u}{2} = 2^e u, \tag{1.25}$$

por tanto

$$| \widetilde{z}_1 - \widetilde{z}_2 | < | \widetilde{z}_1 | \frac{u}{2} \quad \Rightarrow \quad | \widetilde{z}_1 - \widetilde{z}_2 | < 2^e u \quad \Rightarrow \quad \widetilde{z}_1 = \widetilde{z}_2, \tag{1.26}$$

Si $\widetilde{z}_2 \in [2^{e-1}, 2^e)$ entonces la distancia mínima a $\widetilde{z}_1$ sería $2^{e-1} u \leq | \widetilde{z}_1 | \frac{u}{2}$ y por tanto

$$| \widetilde{z}_1 - \widetilde{z}_2 | < | \widetilde{z}_1 | \frac{u}{2} \Rightarrow \quad \widetilde{z}_1 = \widetilde{z}_2, \tag{1.27}$$

con lo que queda demostrado el teorema.

1.5. Comparación de variables

En muchos algoritmos el test de parada incluye el hecho de que dos variables estén próximas entre sí. Habitualmente se cometen dos tipos de errores al comparar números en los algoritmos. El primero de ellos es que en el test de comparación no se tiene en cuenta la magnitud de los números que se comparan y el segundo es que se intenta comparar más allá de lo que permite la calidad de la aritmética y el test de comparación no se cumple hasta que los valores sean exactamente iguales. Para hacer

correctamente una comparación se fija un umbral o tolerancia TOL y expresaremos que las variables A y B están cercanas entre sí con una tolerancia TOL si se cumple cualquiera de las siguientes condiciones:

$$\begin{aligned} &| A - B | \leq (| A | + \epsilon) TOL, \\ &| A - B | \leq (| B | + \epsilon) TOL, \\ &| A - B | \leq (max \{| A |, | B |\} + \epsilon) TOL, \end{aligned}$$

donde $\epsilon > 0$ es un número muy pequeño que se usa para que el test funcione correctamente en los casos en que las variables A ó B sean 0 o muy pequeñas. Por ejemplo, si $B = 0$ y $\epsilon = 0$ el primer test quedaría

$$| A | \leq | A | TOL, \tag{1.28}$$

que nunca se cumpliría salvo que $A = 0$ ó $TOL > 1$. En el caso de que conozcamos a priori el rango de valores de las variables, es decir sabemos por ejemplo que $A, B \in [a, b]$, donde a, b tienen el mismo orden de magnitud, entonces podemos hacer el test de comparación haciendo

$$| A - B | \leq (max \{| a |, | b |\}) TOL. \tag{1.29}$$

Veamos ahora como elegir TOL en la práctica. Primero observamos que :

$$| A - B | < | A | \cdot TOL \quad \Leftrightarrow \quad B \in (A - |A| TOL, A + |A| TOL). \tag{1.30}$$

En la siguiente tabla se muestra como varía el intervalo $(A - |A|TOL, A + |A|TOL)$ al cambiar TOL tomando $A = 12,345,678$

$$\begin{aligned} |A - B| < |A| 10^{-1} \quad &\rightarrow \quad B \in (11,111,110\ ,\ 13,580,246), \\ |A - B| < |A| 10^{-2} \quad &\rightarrow \quad B \in (12,222,221\ ,\ 12,469,135), \\ |A - B| < |A| 10^{-3} \quad &\rightarrow \quad B \in (12,333,332\ ,\ 12,358,024), \\ |A - B| < |A| 10^{-4} \quad &\rightarrow \quad B \in (12,344,443\ ,\ 12,346,913), \\ |A - B| < |A| 10^{-5} \quad &\rightarrow \quad B \in (12,345,555\ ,\ 12,345,801), \\ |A - B| < |A| 10^{-6} \quad &\rightarrow \quad B \in (12,345,666\ ,\ 12,345,690), \\ |A - B| < |A| 10^{-7} \quad &\rightarrow \quad B \in (12,345,677\ ,\ 12,345,679), \\ |A - B| < |A| 10^{-8} \quad &\rightarrow \quad B \in (12,345,678\ ,\ 12,345,678), \end{aligned}$$

observamos que al elegir TOL=10^{-N}, para que se cumpla el criterio de comparación, las N primeras cifras decimales de A y B deben coincidir. Esto nos da un criterio práctico de como elegir TOL=10^{-N} en función de la precisión que queramos. Nótese que de acuerdo con el teorema 3 si $TOL \leq u/2$ entonces B cumple que $| A - B | < | A | \cdot TOL$ solo si $A = B$ y por tanto el test de comparación no tiene sentido pues se convierte en un test de igualdad. De hecho normalmente se toma TOL significativamente mayor que $u/2$ para evitar los errores de redondeo que se producen

en las operaciones que se acumulan en los últimos decimales de los números. Una elección habitual es tomar

$$TOL = \sqrt{u}, \tag{1.31}$$

que aproximadamente indica que el test $\mid A - B \mid < \mid A \mid \cdot TOL$ se cumple cuando la primera mitad de cifras decimales de A y B coinciden.

Un caso especial se produce cuando comparamos valores periódicos. Por ejemplo, si A y B son ángulos que se mueven entre 0 y 2π, hay que tener en cuenta que si A está muy cerca de 0 y B está muy cerca de 2π, entonces, como ángulos, A y B están muy cerca entre sí. Para tener en cuenta esta periodicidad en $[0, 2\pi]$, podemos comparar las variables de la siguiente manera

$$min\left\{\mid A - B \mid, \mid A - B - 2\pi \mid, \mid A - B + 2\pi \mid\right\} < 2\pi \cdot TOL$$

de esta forma tenemos en cuenta la periodicidad y que la magnitud de los números se mueve en $[0, 2\pi]$

1.6. Problemas resueltos

Problema 1 *Representar el número* 0.0703125 *como*

$$0.0703125 = 2^e \left(1 + \sum_{n=1}^{\infty} \frac{a_n}{2^n}\right).$$

Solución: En primer lugar tenemos que encontrar un entero e tal que

$$1 \leq \frac{0.0703125}{2^e} < 2,$$

para $e = -4$ obtenemos

$$\frac{0.0703125}{2^{-4}} = 1.125,$$

ahora tenemos que escribir el número 1.125 como

$$1.125 = 1 + \sum_{n=1}^{\infty} \frac{a_n}{2^n}$$

los a_n se calculan de la siguiente forma

$$1.125 < 1 + \frac{1}{2^1} = 1.5 \Rightarrow a_1 = 0,$$
$$1.125 < 1 + \frac{1}{2^2} = 1.25 \Rightarrow a_2 = 0,$$
$$1.125 = 1 + \frac{1}{2^3} = 1.125 \Rightarrow a_3 = 1,$$

por tanto

$$0.0703125 = 2^{-4}\left(1+\frac{0}{2}+\frac{0}{2^2}+\frac{1}{2^3}\right),$$

en términos binarios, este numero se escribiría con $e=-4$ y la mantisa viene dada por la secuencia $001000,....$

Problema 2 *Calcular los valores positivos mínimo y máximo que puede tomar un número real en una aritmética de precisión finita en función de t, e_{min} y e_{max}.*

Solución: Los valores positivos mínimo y máximo son

$$x_{min} = 2^{e_{min}},$$
$$x_{max} = 2^{e_{max}}\left(1+\sum_{n=1}^{t}\frac{1}{2^n}\right) = 2^{e_{max}}\frac{1-\frac{1}{2^{t+1}}}{\frac{1}{2}} = 2^{e_{max}}\left(2-\frac{1}{2^t}\right).$$

Problema 3 *Dada una aritmética de precisión finita cualquiera, calcular la distancia que hay entre el número 1 y su inmediato superior (es decir el número que va después de 1), y la distancia entre el número 1 y su inmediato inferior.*

Solución: El número 1 en una aritmética de precisión finita se escribe como

$$1 = 2^0\left(1+\frac{0}{2}+\frac{0}{2^2}+...+\frac{0}{2^t}\right),$$

el número inmediato superior a 1 en la aritmética es

$$2^0\left(1+\frac{1}{2^t}\right) = 1+\frac{1}{2^t},$$

y el número inmediato inferior a 1 viene dado por

$$2^{-1}\left(1+\frac{1}{2}+\frac{1}{2^2}+...+\frac{1}{2^t}\right) = 1-\frac{1}{2^{t+1}}.$$

Problema 4 *Se considera una aritmética de 18 bits donde se dedican 1 bit al signo, 9 bits a la mantisa ($t=9$) y 8 bits al exponente ($e_{min}=-126$, $e_{max}=127$). Escribir, si es posible, los siguientes números en esta aritmética:*

- 2, y los números más cercanos a 2 por arriba y por debajo.

  Solución:

$$2 = 2^1\left(1+\frac{0}{2}+\frac{0}{2^2}+...+\frac{0}{2^9}\right),$$
$$\text{siguiente} = 2^1\left(1+\frac{1}{2^9}\right) = 2+\frac{1}{2^8},$$
$$\text{Anterior} = 2^0\left(1+\frac{1}{2}+\frac{1}{2^2}+...+\frac{1}{2^9}\right) = 2-\frac{1}{2^9}.$$

- El cero, el infinito y NaN.

  Solución:

$$0 = 2^{-127}\left(1+\frac{0}{2}+\frac{0}{2^2}+...+\frac{0}{2^9}\right),$$
$$\infty = 2^{128}\left(1+\frac{0}{2}+\frac{0}{2^2}+...+\frac{0}{2^9}\right),$$
$$NaN = 2^{128}\left(1+\frac{1}{2}+\frac{0}{2^2}+...+\frac{0}{2^9}\right).$$

- Los números positivos más grande y más pequeño de la aritmética (teniendo en cuenta las excepciones)

  Solución:
  Mayor $= 2^{127}\left(\sum_{i=0}^{9}\frac{1}{2^i}\right) = 2^{127}\left(\frac{1-\frac{1}{2^{10}}}{\frac{1}{2}}\right)$,

  Menor (sin tener en cuenta las excepciones) $= 2^{-126}$,
  Menor (teniendo en cuenta las excepciones) $= 2^{-127}\left(\frac{1}{2^9}\right)$.

- $\frac{1}{9}$.

  Solución: No se puede escribir de forma exacta. Si suponemos

$$\frac{1}{9} = 2^e\left(1+\sum_{i=1}^{t}\frac{a_i}{2^i}\right) \Longrightarrow 1 = 9\cdot 2^e\left(1+\sum_{i=1}^{t}\frac{a_i}{2^i}\right) \Longrightarrow$$
$$\Longrightarrow 2^{t-e} = 3^2\left(1+\sum_{i=1}^{t}a_i 2^{t-i}\right) \Longrightarrow 2^{t-e} = 3^2 m,$$

  donde m es un número entero. Ahora bien esta igualdad es imposible porque resultaría que 3 divide a 2.

- $2\left(1-\frac{1}{2^9}\right)$.

Solución: $2\left(1-\frac{1}{2^9}\right) = 2^0\left(1+\frac{1}{2}+\frac{1}{2^2}+\frac{1}{2^3}+\frac{1}{2^4}+\frac{1}{2^5}+\frac{1}{2^6}+\frac{1}{2^7}+\frac{1}{2^8}\right)$

Problema 5 *Dado un número* $\widetilde{z} = 2^e\left(1+\sum_{n=1}^{t}\frac{a_n}{2^n}\right)$ *en una aritmética de precisión finita. Calcular el número inmediatamente inferior y superior a él en dicha aritmética.*

Solución: Sea $u = 2^{-t}$ la unidad de redondeo. Se tiene que en cada intervalo $[2^e, 2^{e+1})$ los números de la aritmética están a distancia $2^e u$. Por tanto, si el número es de la forma

$$\widetilde{z} = 2^e,$$

entonces el inmediato superior es

$$\widetilde{z} + 2^e u,$$

y el inmediato inferior que se encuentra en $[2^{e-1}, 2^e)$ es

$$\widetilde{z} - 2^{e-1} u,$$

para cualquier otro número $\widetilde{z}$, el inmediato superior e inferior son

$$\widetilde{z} + 2^e u \qquad \text{y} \qquad \widetilde{z} - 2^e u$$

Problema 6 *Calcular las raíces del polinomio* $P(x) = x^2 - 2x + 0.01$ *evitando los errores de cancelación.*

Solución:

$$x_1 = \frac{2 + \sqrt{4 - 0.04}}{2} = 1.995, \qquad x_2 = \frac{c}{x_1 a} = \frac{0.01}{1.995}.$$

Problema 7 *Describir un algoritmo para calcular el exponente mínimo,* e_{min}*, de una aritmética de precisión finita*

Solución: tendremos en cuenta que se debe cumplir

$$2^{e_{min}} > 0 \quad y \quad \frac{2^{e_{min}}}{2} = 0,$$

es decir, al dividir por 2 el número positivo más pequeño, el resultado se almacena como 0 en la aritmética. Por tanto, para calcular el exponente mínimo, e_{min}, basta hacer un bucle donde se divide por dos sucesivamente un número que se inicializa a 1. Las iteraciones paran cuando el resultado de la división es cero.

Problema 8 *Describir un algoritmo para calcular el exponente máximo,* e_{max}*, de una aritmética de precisión finita*

Solución: tendremos en cuenta que el número mayor de la aritmética $\widetilde{z}_{max} = 2^{e_{max}} \left(2 - \frac{1}{2^t}\right)$ debe cumplir

$$2 \cdot \widetilde{z}_{max} = \widetilde{z}_{max},$$

es decir, al multiplicar por 2 el número $\widetilde{z}_{max}$, el resultado se almacena como el mismo número en la aritmética. Por tanto, para calcular el exponente máximo, e_{max}, basta hacer un bucle donde se multiplica por dos sucesivamente un número que se inicializa a 1. Las iteraciones paran cuando al multiplicar por dos el número no cambia.

Problema 9 *Describir un algoritmo para calcular la precisión de la aritmética,* t*, de una aritmética de precisión finita*

Solución: tendremos en cuenta que la unidad de redondeo $u = 2^{-t}$ satisface:

$$1 + u > 1 \quad y \quad 1 + \frac{u}{2} = 1,$$

por tanto, para calcular la precisión de la aritmética, t, basta hacer un bucle donde se va dividiendo por dos un número que se inicializa a 1, y a este número se le suma 1, cuando el número resultado sea igual a 1 salimos del bucle.

Problema 10 *Sea $A \neq 0$. Como hay que elegir TOL para que el test*

$$|A - B| \leq |A|TOL,$$

se cumpla cuando todos los decimales de A y B coincidan salvo los M últimos.

Solución: Sea u la unidad de redondeo de la aritmética. Sabemos que u determina los decimales que admite la aritmética. Por otro lado el test

$$|A - B| \leq |A|10^{-N}$$

se cumple si los primeros N decimales de A y B coinciden. Por tanto tendremos que elegir $TOL \approx u10^M$ para que los decimales de A y B coincidan salvo los M últimos.

Problema 11 *En una aritmética de 32 bits calcular el resultado de la asignación*

$$A = 2^{16} + \frac{1}{2^2} + \frac{1}{2^{10}}$$

determinando el exponente y la mantisa del resultado en forma binaria

Solución: Teniendo en cuenta que

$$2^{16} + \frac{1}{2^2} + \frac{1}{2^{10}} = 2^{16}\left(1 + \frac{1}{2^{18}} + \frac{1}{2^{26}}\right)$$

como en la aritmética de 32 bits $t = 23$, a lo más que puede llegar la mantisa es a $\frac{1}{2^{23}}$, por tanto el resultado de la asignación sería

$$A = 2^{16}\left(1 + \frac{1}{2^{18}}\right)$$

es decir sería el número de exponente $e = 16$ y mantisa en forma binaria

$$00000000000000000100000$$

donde el 1 está en la posición 18.

Problema 12 *Calcular cuantos números positivos distintos tiene una aritmética en función de e_{min}, e_{max}, y la precisión t.*

Solución: para cada exponente $e \in Z$ con $e_{min} \leq e \leq e_{max}$ hay 2^t números positivos. Por tanto el total de números positivos es

$$(e_{max} - e_{min} + 1)2^t$$

Problema 13 *Estudiar si en la aritmética estándar de* 32 *bits todos los números naturales entre 1 y 10,000,000 están representados en la aritmética. ¿y entre 1 y 100,000,000?*

Solución: Dado un exponente $e \in Z$, los números de la aritmética se reparten uniformemente en el intervalo $[2^e, 2^{e+1})$ y están a una distancia entre ellos de $2^e u$. donde $u = 2^{-t}$, por tanto todos los números naturales estarán mientras se cumpla que $2^e u \leq 1$. En particular para el intervalo $[2^t, 2^{t+1})$ la distancia entre los puntos de la aritmética es 1, y por tanto a partir de 2^{t+1} es cuando las distancia entre los puntos es mayor que 1 y por tanto no están todos los números naturales. Como $t = 23$ en una aritmética de 32 bits, se tiene que $2^{t+1} = 2^{24} = 16,777,216$, lo cual significa que todos los naturales entre 1 y 16,777,216 están en la aritmética y a partir de $16,777,216$ ya no están todos los que siguen pues empiezan a separarse entre ellos primero de 2 en 2, después de 4 en 4, etc.. Por tanto todos los números naturales entre 1 y 10,000,000 están pero no todos los que están entre 1 y 100,000,000.

Capítulo 2

CÁLCULO DE LOS CEROS DE UNA FUNCIÓN

En este tema vamos a estudiar algunos métodos numéricos para aproximar los ceros de una función de una variable, $f(x)$, esto es, los valores de x para los cuales $f(x) = 0$. Este problema es de gran importancia en múltiples aplicaciones donde hay que resolver ecuaciones para las cuales no hay un procedimiento matemático que permita calcular la solución exacta. A continuación se muestran algunas ecuaciones de este tipo en diferentes ámbitos:

- La ecuación de Kepler en Astronomía

$$f(x) = x - A\sin(x) - B = 0, \tag{2.1}$$

- Ecuación que aparece en hidraúlica de conducciones

$$f(x) = \tan(x) - x = 0, \tag{2.2}$$

- Alcance de un proyectil en tiro parabólico con rozamiento

$$f(x) = Ax + B\ln(1 - Cx) = 0, \tag{2.3}$$

- Cálculo del interés aplicado a un plan de pensiones

$$f(x) = \sum_{n=0}^{N} A(1. + x)^{N-n} - B = 0, \tag{2.4}$$

donde A, B, C son parámetros de los respectivos modelos. Todas estas ecuaciones tienen en común que no existe un procedimiento para calcular su solución exacta y por tanto hay que diseñar métodos numéricos que permitan aproximar la solución. De hecho, con frecuencia, la función $f(x)$ no se evalúa a través de una fórmula sino a través de un algoritmo complejo. Por ejemplo, en el caso de que $f(x)$ sea una

estimación de donde va a aterrizar una nave espacial en función de la posición, x, sobre la órbita, en que abandona su trayectoria orbital, se tiene que $f(x)$ no se calcula usando una única fórmula matemática como en los ejemplos anteriores, se calcula usando un algoritmo complejo que usa un gran número de variables. Si se quiere que la nave espacial aterrice donde están preparados para recoger a los astronautas es necesario resolver un problema del tipo $f(x) = y$, donde y representa las coordenadas del punto de aterrizaje. Nótese que el problema $f(x) = y$ se puede plantear como un problema de cálculo de ceros tomando simplemente la función $\tilde{f}(x) = f(x) - y = 0$.

Otra situación práctica donde aparece el cálculo de ceros es en los problemas de optimización donde se quiere calcular los máximos o mínimos de una función $F(x)$, en ese caso, se usa que los máximos/mínimos corresponden a valores donde la derivada es 0 y por tanto se resuelve el problema $F'(x) = 0$.

A continuación presentaremos diversos métodos numéricos para el cálculo de ceros de una función.

2.1. Método de la bisección.

Se considera una función continua $f(x)$ y un intervalo $[a, b]$ donde la función $f(x)$ cambia de signo, es decir $f(a) \cdot f(b) < 0$. El método consiste en dividir el intervalo $[a, b]$ por la mitad en 2 subintervalos y a continuación $[a, b]$ se actualiza al subintervalo donde se produce cambio de signo de la función. Este proceso se repite iterativamente. El punto medio del intervalo se calcula haciendo $x = \frac{a+b}{2}$. Las iteraciones se paran si $f(\frac{a+b}{2}) = 0$ o si a y b son próximos entre sí módulo una tolerancia TOL.

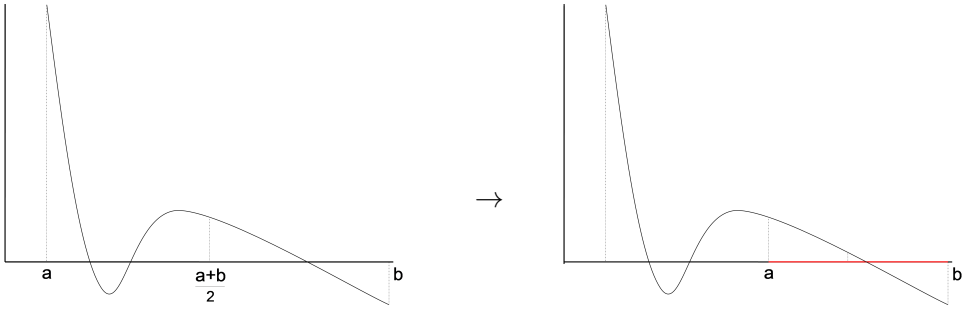


Ilustración gráfica de una iteración del método de la bisección.

Descripción del algoritmo del método de la bisección.

- Se parte de una función $f(x)$ y de un intervalo $[a, b]$ donde la función cambia de signo.
- Se inicia un procedimiento iterativo donde se va actualizando el intervalo $[a, b]$ partiendo el intervalo en dos mitades y quedándonos con la mitad que mantiene el cambio de signo.
- Las iteraciones paran cuando la función vale cero en el punto medio del intervalo o cuando el tamaño del intervalo es muy pequeño módulo una tolerancia TOL.

En cualquier algoritmo iterativo siempre hay que plantearse si el algoritmo puede entrar en un bucle infinito, en cuyo caso hay que añadir un parámetro adicional que sería el número máximo de iteraciones permitidas. El método de la bisección es de los escasos algoritmos donde podemos asegurar que no va a entrar en un bucle infinito debido a que en cada iteración dividimos por 2 el tamaño del intervalo $[a, b]$, por tanto, al trabajar con una aritmética de precisión finita, en un número finito de iteraciones se va a cumplir que $a = b$ y por tanto el algoritmo pararía. De hecho, aunque el método de la bisección es, en general, más lento que el resto de métodos que vamos a estudiar, es el único que converge seguro, aunque tiene la limitación de que necesita partir de un intervalo donde la función cambie de signo, lo cual en ocasiones puede ser difícil de encontrar.

2.2. Método de la regula-falsi

Este método es una variación del anterior en el sentido siguiente: en lugar de tomar el punto medio $\frac{a+b}{2}$ para dividir el intervalo, se considera el punto de intersección de la recta que pasa por los puntos $(a, f(a))$ y $(b, f(b))$ con el eje x. Dicha recta viene dada por la ecuación

$$\frac{y - f(a)}{f(b) - f(a)} = \frac{x - a}{b - a}, \tag{2.5}$$

por tanto, el punto que se usa para dividir el intervalo es

$$x_r = a - \frac{b - a}{f(b) - f(a)} f(a), \tag{2.6}$$

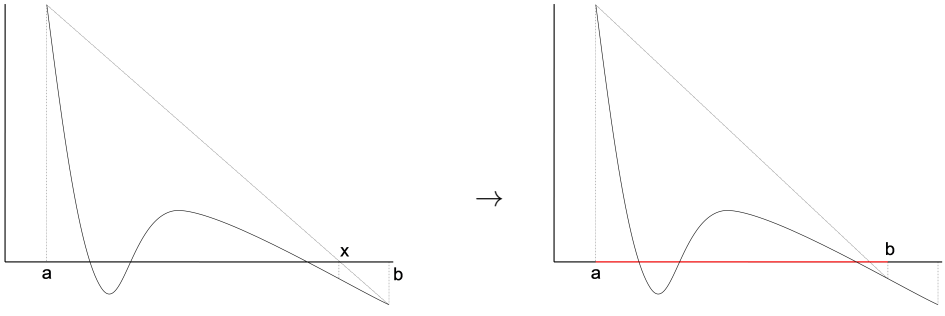


Ilustración gráfica de una iteración del método de la regula-falsi.

Descripción del algoritmo de la regula-falsi.

- Se parte de una función $f(x)$ y de un intervalo $[a, b]$ donde la función cambia de signo.
- Se inicia un procedimiento iterativo donde se va actualizando el intervalo $[a, b]$ dividiendo el intervalo en dos trozos usando como punto de corte x_r (ecuación (2.6)) y quedándonos con el trozo que mantiene el cambio de signo.
- Las iteraciones paran cuando la función vale cero en el punto x_r o cuando el nuevo x_r está muy próximo al calculado en la iteración anterior módulo una tolerancia TOL o cuando se excede el número de iteraciones máximo.

El método de la regula-falsi es, en general, más rápido que el método de la bisección porque normalmente el cero de $f(x)$ en $[a, b]$ está más cerca del extremo del intervalo donde la función sea más pequeña. Sin embargo no podemos asegurar que esto sea siempre así e incluso en casos excepcionales podría darse que el algoritmo no converge. Por ello en este algoritmo resulta apropiado controlar el máximo número de iteraciones con un parámetro adicional.

2.3. Método de Newton-Raphson

Éste es, sin duda, uno de los métodos más importantes y útiles para el cálculo de raíces. A diferencia de los métodos anteriores no necesita de un intervalo inicial donde la función cambie de signo. Dada una aproximación inicial de la raíz x_0, se busca, a partir de x_0, una aproximación mejor x_1 de la raíz, de la siguiente forma: Se sustituye la función $f(x)$ por el valor de su desarrollo de Taylor centrado en x_0 hasta el orden 1, es decir

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0), \tag{2.7}$$

que corresponde a un polinomio de grado 1, y a continuación se calcula x_1 como el cero de este polinomio, es decir:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}, \tag{2.8}$$

de forma iterativa, obtenemos, a partir de x_0 una secuencia x_n de valores que van aproximando la raíz, definidos por

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \tag{2.9}$$

en la práctica, en memoria no hace falta guardar la secuencia entera x_n, es suficiente con almacenar solo x_0 y x_1 y se van actualizando los valores en cada iteración, de tal forma que x_1 representa el valor calculado en la última iteración realizada.

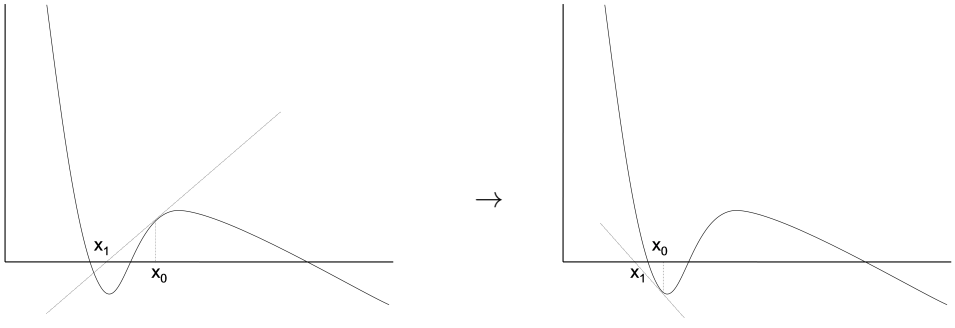


Ilustración gráfica del método de Newton-Raphson.

Descripción del algoritmo del método de Newton-Raphson.

- Se parte de una función $f(x)$ y de una aproximación inicial x_0 de la raíz.
- Se inicia un procedimiento iterativo para ir actualizando x_0 utilizando el nuevo valor x_1 dado por la ecuación (2.8).
- las iteraciones paran si la función vale 0 en x_0, si la derivada de la función vale 0 en x_0, si la distancia entre x_1 y x_0 es pequeña módulo una tolerancia TOL o cuando se excede el número de iteraciones máximo.

Este algoritmo no converge siempre, especialmente en los casos en que la aproximación inicial x_0 está muy alejada de un cero de la función y por tanto requiere controlar el máximo número de iteraciones con un parámetro adicional.

2.4. Método de la secante

Este método es una variante del método de Newton-Raphson para el caso en que la función derivada de $f(x)$ no esté implementada en el código. Hay que tener en cuenta que a nivel de código de programación para acceder a $f'(x)$ tendríamos que tener acceso a una implementación de la derivada lo cual no sucede siempre. En ese caso nos apoyamos en dos aproximaciones iniciales x_0, x_1 y a partir de estos dos se calcula x_2, el desarrollo de Taylor se centra en x_1 y se se sustituye el valor $f'(x_1)$ en el algoritmo, por una aproximación dada por

$$f'(x_1) \approx \frac{f(x_1) - f(x_0)}{x_1 - x_0}, \tag{2.10}$$

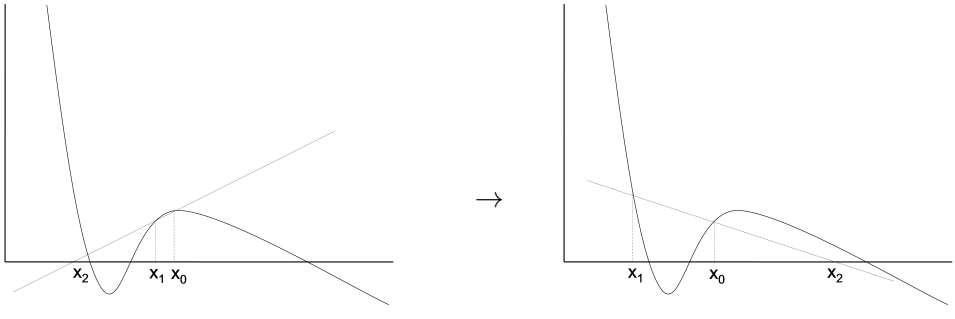


Ilustración gráfica del método de la secante

Descripción del algoritmo del método de la secante

- Se parte de una función $f(x)$ y de dos aproximaciones iniciales x_0 y x_1 de la raíz.
- Se inicia un procedimiento iterativo para ir actualizando x_0 y x_1 utilizando el nuevo valor x_2 dado por la ecuación

$$x_2 = x_1 - f(x_1)\frac{x_1 - x_0}{f(x_1) - f(x_0)}$$

- las iteraciones paran si la función vale 0 en x_1, si la función vale lo mismo en x_0 y en x_1, si la distancia entre x_2 y x_1 es pequeña módulo una tolerancia TOL o cuando se excede el número de iteraciones máximo.

Este algoritmo requiere controlar el máximo número de iteraciones con un parámetro adicional.

2.5. Método de Müller.

Este método es una generalización del método de Newton-Raphson. En lugar de quedarnos con la parte lineal del desarrollo de Taylor de la función, nos quedamos con los términos hasta el orden 2, de tal forma que hacemos

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2, \tag{2.11}$$

donde x_0 es una aproximación de una raíz de la función $f(x)$. El polinomio anterior representa la parábola tangente en el punto x_0. Para obtener una aproximación x_1 mejor de la raíz calculamos los ceros del polinomio de segundo grado anterior, es decir

$$x_1 = x_0 + \frac{-f'(x_0) \pm \sqrt{f'(x_0)^2 - 2f(x_0)f''(x_0)}}{f''(x_0)}, \tag{2.12}$$

de las dos posibles raíces, nos quedamos con aquella que esté más cercana a x_0. Este proceso se repite iterativamente hasta converger.

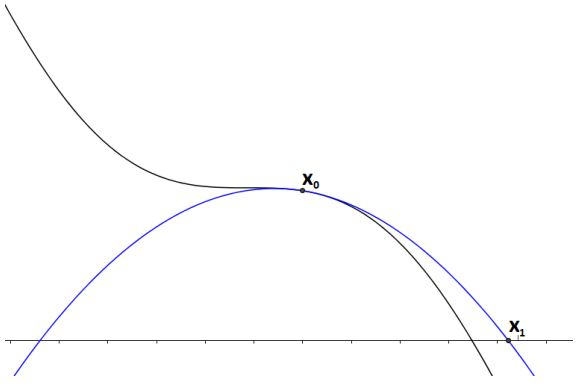


Ilustración gráfica del método de Müller.

Descripción del algoritmo del método de Müller

- Se parte de una función $f(x)$ y de una aproximación inicial x_0 de la raíz.
- Se inicia un procedimiento iterativo para ir actualizando x_0 utilizando el nuevo valor x_1 la raíz más cercana a x_0 del polinomio dado por la ecuación (2.11).
- Las iteraciones paran si la función vale 0 en x_0, si el polinomio de la ecuación (2.11) no tiene raíces realeas, si la distancia entre x_1 y x_0 es pequeña módulo una tolerancia TOL o cuando se excede el número de iteraciones máximo.
- Para calcular las raíces de un polinomio de grado dos se utilizará una función previamente implementada.

Este algoritmo requiere controlar el máximo número de iteraciones con un parámetro adicional.

En el caso en que al implementar el algoritmo no se tenga acceso a funciones que calculen la primera y segunda derivada de $f(x)$, necesitamos aproximar dichas derivadas. Como veremos más adelante, se pueden usar las siguientes aproximaciones

$$f'(x_0) \approx \frac{f(x_0+h)-f(x_0-h)}{2h} \quad f''(x_0) \approx \frac{f(x_0-h)-2f(x_0)+f(x_0+h)}{h^2}. \quad (2.13)$$

donde h se calcula en función de la calidad de la aritmética. Si u es la unidad de redondeo de la aritmética, podemos usar

$$h = (|x_0| + 1)\sqrt{u}$$

de esta manera, en general, los valores $x_0 + h$ y $x_0 - h$, coinciden con x_0, apróximadamente, en la primera mitad de los bits de la mantisa.

2.6. Cálculo de las raíces de un polinomio.

Los polinomios son un tipo particular de funciones que, por su gran utilidad, requieren un análisis algo más detallado. Nos ocuparemos sólo de las raíces reales de los polinomios, aunque también hay que indicar que existen algoritmos que pueden calcular raíces complejas. Existen fórmulas para calcular las raíces de polinomios hasta grado 4. Para grados mayores, el matemático francés Galois demostró que no existe una fórmula general para calcular las raíces de polinomios de grado mayor que 4 en función de sus coeficientes.

En la práctica, los polinomios son de gran importancia porque hay muchas funciones que se aproximan por polinomios usando los desarrollos de Taylor u o otras técnicas y en algunas aplicaciones, como las matemáticas financieras, aparecen con frecuencia polinomios. Por ejemplo como se vio al principio de este tema para calcular el interés de un plan de pensiones hay que resolver una ecuación del tipo

$$f(x) = \sum_{n=0}^{N} A(1+x)^{N-n} - B = 0, \quad (2.14)$$

que es un polinomio en x. En general, un polinomio se escribe como $P(x) = a_0 + a_1x + ... + a_{n-1}x^{n-1} + a_nx^n$. Los coeficientes $\{a_k\}$ se almacenan en el ordenador como un vector de tamaño $n + 1$. Un sencillo algoritmo para evaluar un polinomio en un punto x sería

Descripción algoritmo para evaluar un polinomio de grado n en x. La evaluación se guarda en la variable P_x

- si inicializa $P_x = a_0$ y una variable $xk = x$ donde en cada iteración guardamos x^k.
- hacemos un proceso iterativo desde 1 hasta n y vamos actualizando P_x y xk de acuerdo con la ecuación del polinomio.

Como veremos a continuación existen otras formas más eficientes de evaluar un polinomio.

Algoritmo de Horner para evaluar un polinomio en un punto

Dado un polinomio $P(x) = a_0 + a_1x + ... + a_{n-1}x^{n-1} + a_nx_0^n$, es posible compactarlo de la forma siguiente:

$$P(x) = a_0 + x\left(a_1 + x\left(a_2 + x\left(a_3 + x(..... + x\left(a_{n-1} + xa_n\right)\right)\right)\right)\right). \tag{2.15}$$

El siguiente resultado muestra una forma rápida y sencilla de evaluar simultáneamente un polinomio y su derivada usando la expresión anterior.

Teorema 4 *(Método de Horner). Sea* $P(x) = a_nx^n + a_{n-1}x^{n-1} + ... + a_0$. *Dado* x_0, $P(x)$ *se puede descomponer como*

$$P(x) = Q(x)(x - x_0) + P(x_0) \tag{2.16}$$

donde $Q(x)$ *es un polinomio de grado* $n-1$, *además, se verifica que* $P'(x_0) = Q(x_0)$ *y si definimos iterativamente, de forma descendente* $\{b_k\}$ *y* $\{c_k\}$ *como*

$$b_k = a_k + b_{k+1}x_0 \quad con \quad b_n = a_n,\ k = n-1, .., 0, \tag{2.17}$$

$$c_k = b_k + c_{k+1}x_0 \quad con \quad c_n = a_n,\ k = n-1, .., 1 \tag{2.18}$$

entonces se verifica que

$$P(x_0) = b_0, \tag{2.19}$$

$$P'(x_0) = Q(x_0) = c_1. \tag{2.20}$$

Este teorema permite evaluar el polinomio y su derivada en un punto de forma muy sencilla, como muestra el siguiente algoritmo

Descripción algoritmo para evaluar un polinomio $P(x)$ de grado n y su derivada $P'(x)$ usando el algoritmo de Horner. La evaluación se guarda en las variables b=$P(x)$ y c=$P'(x)$.

- Se inicializa $b = a_n$ y $c = a_n$
- Hacemos un proceso iterativo desde n-1 hasta 1 y vamos actualizando b y c de acuerdo con las ecuaciones (2.17) y (2.18).
- Al terminar el bucle actualizamos b usando la ecuación (2.17) con $k = 0$.

Nótese que en las variables b y c se están guardando en cada momento los valores de $b[k]$ y $c[k]$ que se van actualizando en cada iteración. El siguiente resultado determina un intervalo donde deben estar todas las raíces del polinomio:

Teorema 5 *Sea un polinomio $P(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_0$ con $a_n \neq 0$, entonces las raíces reales de $P(x)$ están en el intervalo $[-P_{max}, P_{max}]$ donde P_{max} viene dado por*

$$P_{max} = 1 + \frac{max_{k=0,..,n-1} \mid a_k \mid}{\mid a_n \mid}. \tag{2.21}$$

Demostración Veamos que si $|x| > P_{max}$, entonces $|P(x)| > 0$. Efectivamente,

$$\begin{aligned} |P(x)| &\geq |a_n x^n| - max_{k=0,..,n-1} |a_k| \sum_{k=0}^{n-1} |x|^k = \\ &= |a_n| \, |x|^n - max_{k=0,..,n-1} |a_k| \, \frac{1 - |x|^n}{1 - |x|} \geq \\ &\geq |a_n| \, |x|^n - max_{k=0..,n-1} |a_k| \, \frac{|x|^n}{|x| - 1} = \\ &= \frac{|x|^n \left(|a_n| \left(|x| - 1 \right) - max_{k=0,..,n-1} |a_k| \right)}{|x| - 1} > 0. \end{aligned}$$

Teorema 6 *Entre dos raíces de una función derivable $f(x)$ hay una raíz de $f'(x)$.*

Demostración: este resultado es el conocido teorema de Rolle.

Teorema 7 *La derivada $P'(x)$ del polinomio $P(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_0$ es*

$$P'(x) = n a_n x^{n-1} + (n-1) a_{n-1} x^{n-2} + ... + 2a_2 x + a_1. \tag{2.22}$$

Demostración. Es inmediata derivando el polinomio $P(x)$.

La combinación de los dos resultados anteriores permite aislar las posibles raíces de $P(x)$ de la forma siguiente: las m raíces distintas $x_1 < x_2 < .... < x_m$ de $P(x)$ (con $m \leq n$) están intercaladas en el intervalo $[-P_{max}, P_{max}]$ con las raíces $x'_1 < x'_2 < .... < x'_{m-1}$ de $P'(x)$, es decir

$$-P_{max} \leq x_1 \leq x'_1 \leq x_2 \leq x'_2 \leq ... \leq x'_{m-1} \leq x_m \leq P_{max}, \tag{2.23}$$

Volviendo a aplicar este razonamiento sucesivamente sobre $P'(x)$, $P''(x)$, etc. para intercalar los ceros de una derivada con los ceros de la siguiente, podemos deducir el siguiente algoritmo para calcular todas las raíces de un polinomio $P(x)$:

Decripción algoritmo para el cálculo de las raíces reales de un polinomio y de sus polinomios derivadas.

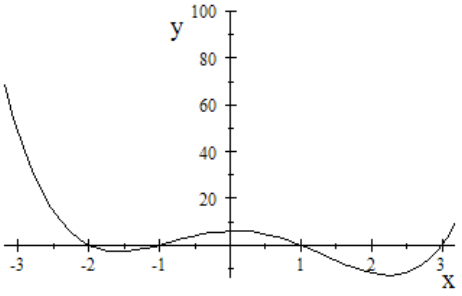
1. Se implementa un procedimiento que calcula las raíces de un polinomio a partir de las raíces de su polinomio derivada $x'_0 < x'_1 < \cdots < x'_n$. Para ello se construye un vector con todos los posibles extremos de intervalos donde puede haber raíces del polinomio. Este vector se construye añadiendo al vector de ceros de la derivada el valor $-P_{max}$ por la izquierda y el valor P_{max} por la derecha. A continuación se van añadiendo a un vector los ceros que van saliendo tomando todos los intervalos posibles de izquierda a derecha, teniendo en cuenta que para que exista un cero en un intervalo tiene que haber cambio de signo en los extremos.

2. Se realiza un proceso iterativo donde empezando por la deriva $n-1$ del polinomio se va hacia atrás calculando los ceros de cada polinomio derivada usando los ceros de la derivada siguiente calculados en la iteración anterior. (Se parte de un vector de ceros vacío que se va actualizando en cada iteración).

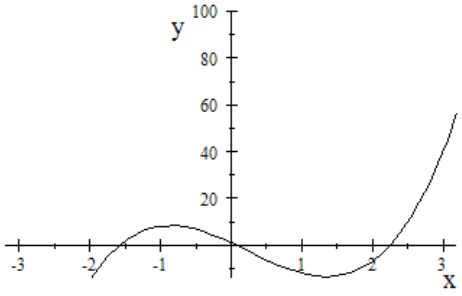
3. Cuando termina el proceso iterativo, tendremos los ceros del polinomio original

El algoritmo anterior es relativamente sencillo y permite calcular todas las raíces reales de los polinomios. Existen métodos que permiten calcular las raíces complejas de los polinomios, pero que utilizan técnicas más complejas que no se estudiarán en este libro.

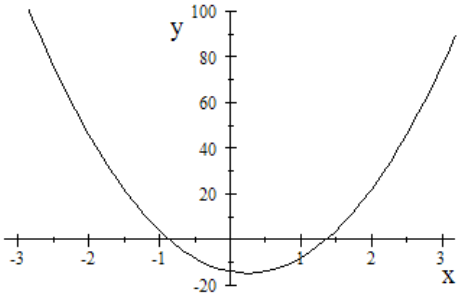
Ejemplo 2 *Consideremos el polinomio* $P(x) = x^4 - x^3 - 7x^2 + x + 6$. *A continuación se muestran las gráficas del polinomio y sus derivadas:*



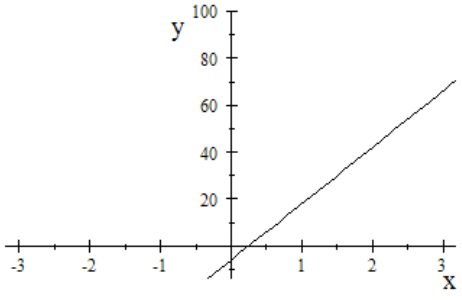
$P^{0)}(x) = x^4 - x^3 - 7x^2 + x + 6$



$P^{1)}(x) = 4x^3 - 3x^2 - 14x + 1.$



$P^{2)}(x) = 12x^2 - 6x - 14.$



$P^{3)}(x) = 24x - 6.$

El método funcionaría de la siguiente forma:

1. Calculamos las derivadas del polinomio y los extremos de los intervalos donde deben tener sus ceros (usando la fórmula ((2.21)).

   - $P^{0)}(x) = x^4 - x^3 - 7x^2 + x + 6 \quad \rightarrow \quad P^{0)}_{max} = 1 + \frac{7}{1} = 8,$
   - $P^{1)}(x) = 4x^3 - 3x^2 - 14x + 1 \quad \rightarrow \quad P^{1)}_{max} = 1 + \frac{14}{4} = 4.5,$
   - $P^{2)}(x) = 12x^2 - 6x - 14 \quad \rightarrow \quad P^{2)}_{max} = 1 + \frac{14}{12} = 2.166,$
   - $P^{3)}(x) = 24x - 6,$

2. Calculamos el cero de la derivada $P^{3)}(x)$, es decir $x_0^{3)} = \frac{6}{24} = 0.25$.

3. Calculamos los ceros de la derivada $P^{2)}(x)$ en los intervalos $[-P^{2)}_{max}, x_0^{3)}]$ y $[x_0^{3)}, P^{2)}_{max}]$. Observamos que hay cambio de signo en los dos intervalos y por tanto hay 2 raíces que calculamos por el método de la bisección en cada intervalo y obtenemos $x_0^{2)} = -0.858$ y $x_1^{2)} = 1.358$.

4. Calculamos los ceros de la derivada $P^{1)}(x)$ en los intervalos $[-P^{1)}_{max}, x_0^{2)}]$, $[x_0^{2)}, x_1^{2)}]$ y $[x_1^{2)}, P^{1)}_{max}]$. Observamos que hay cambio de signo en los tres intervalos y por tanto hay 3 raíces que calculamos por el método de la bisección en cada intervalo y obtenemos $x_0^{1)} = -1.574$, $x_1^{1)} = 7.05 \times 10^{-2}$ y $x_2^{1)} = 2.253$.

5. Calculamos los ceros $P^{0)}$ en los intervalos $[-P^{0)}_{max}, x^{1)}_0]$, $[x^{1)}_0, x^{1)}_1]$, $[x^{1)}_1, x^{1)}_2]$ y $[x^{1)}_2, P^{2)}_{max}]$. Observamos que hay cambio de signo en los cuatro intervalos y por tanto hay 4 raíces que calculamos por el método de la bisección en cada intervalo y obtenemos $x^{0)}_0 = -2, x^{0)}_1 = -1, x^{0)}_2 = 1, x^{0)}_3 = 3$.

2.7. Problemas resueltos

Problema 14 *Dar un ejemplo de función $f(x)$ e intervalo $[a, b]$ donde la función cambie de signo pero $f(x) = 0$ no tenga ninguna solución en el intervalo. ¿En ese caso, hacía donde converge el método de la bisección?*

Solución: Tienen que ser funciones discontinuas en $[a, b]$. Por ejemplo que tengan una asíntota. $f(x) = tan(x)$ en intervalo $[\frac{\pi}{4}, \frac{3\pi}{4}]$ cumple estas condiciones. En estos casos el método de la bisección converge hacia un valor x_0 donde $f(x)$ sea discontinua y $f(x)$ cambie de signo a la izquierda y derecha de x_0.

Problema 15 *Calcular 2 iteraciones del algoritmo de la bisección para buscar un cero de la función $f(x) = x^2 - 2$ en el intervalo $[a, b] = [-2, 0]$.*

Solución: primero observamos que hay cambio de signo de la función en los extremos del intervalo: $f(-2) > 0$ y $f(0) < 0$.

Iteración 1

$$x_m = \frac{0 + (-2)}{2} = -1 \quad \rightarrow \ f(-1) < 0 \quad \rightarrow \ [a, b] = [-2, -1].$$

Iteración 2

$$x_m = \frac{-1 + (-2)}{2} = -1.5 \quad \rightarrow \ f(-1.5) > 0 \quad \rightarrow \ [a, b] = [-1.5, -1].$$

Problema 16 *Calcular 2 iteraciones del algoritmo de la regula-falsi para buscar un cero de la función $f(x) = x^2 - 2$ en el intervalo $[a, b] = [0, 2]$.*

Solución: primero observamos que hay cambio de signo en el intervalo, es decir, $f(0) < 0$ y $f(2) > 0$. A continuación aplicamos la fórmula general para calcular el punto usado para dividir el intervalo:

$$x_r = a - \frac{b - a}{f(b) - f(a)} f(a), \tag{2.24}$$

Iteración 1

$$x_r = 0 - \frac{2}{f(2) - f(0)} f(0) = 1 \quad \rightarrow \ f(1) < 0 \quad \rightarrow \ [a, b] = [1, 2].$$

Iteración 2

$$x_r = 1 - \frac{1}{f(2) - f(1)} f(1) = \frac{4}{3} \quad \rightarrow \ f(\frac{4}{3}) < 0 \quad \rightarrow \ [a, b] = [\frac{4}{3}, 2].$$

Problema 17 *Calcular dos iteraciones del método de Newton-Raphson para calcular un cero de la función* $f(x) = x^3 - 3$ *partiendo de* $x_0 = 1$.

Solución: Aplicamos la fórmula general

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

y obtenemos

$$\text{Iteración 1} \qquad x_1 = 1 - \frac{-2}{3} = \frac{5}{3}.$$

$$\text{Iteración 2} \qquad x_2 = \frac{5}{3} - \frac{(\frac{5}{3})^3 - 3}{3(\frac{5}{3})^2} = 1.47111.$$

Problema 18 *Calcular una iteración del método de la secante para calcular un cero de la función* $f(x) = x^3 - 3$ *partiendo de* $x_0 = 0$, $x_1 = 1$.

Solución: Aplicamos la fórmula general

$$x_{n+1} = x_n - \frac{f(x_n)}{\frac{f(x_n)-f(x_{n-1})}{x_n - x_{n-1}}},$$

y obtenemos

$$x_2 = x_1 - \frac{f(x_1)}{\frac{f(x_1)-f(x_0)}{x_1-x_0}} = 1 - \frac{-2}{\frac{-2-(-3)}{1-0}} = 3.$$

Problema 19 *Calcular una iteración del método de Müller para calcular un cero de la función* $f(x) = x^3 - 3$ *partiendo de* $x_0 = 1$ *(Calcular las derivadas de forma exacta).*

Solución: Usando el desarrollo de Taylor hasta el orden 2 obtenemos

$$f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 = -2 + 3(x-1) + 3(x-1)^2 = 0,$$

las raíces de este polinomio son

$$x = 1 + \frac{-3 \pm \sqrt{33}}{6},$$

por tanto, tomando la raíz más cercana a 1 nos queda:

$$x_1 = 1 + \frac{-3 + \sqrt{33}}{6} = 1.45743.$$

Problema 20 *Dado el polinomio* $P(x) = 2x^3 + 3x^2 + 4x + 5$. *Evaluar el polinomio y su derivada en el punto* $x_0 = 2$, *utilizando el algoritmo de Horner.*

Solución: compactamos el polinomio para aplicar el algoritmo de Horner.

$$P(x) = ((2x+3)x+4)x+5 \quad \rightarrow \quad P(2) = ((7)2+4)2+5 = (18)2+5 = 41,$$

además, aplicando el algoritmo que se deduce del teorema de Horner visto anteriormente se tiene que

$$P'(x_0) = Q(x_0) = (2x_0+7)x_0+18 \quad \rightarrow \quad P'(2) = (4+7)2+18 = 40.$$

Problema 21 *Aislar en intervalos las raíces del polinomio*

$$P(x) = 20x^3 - 45x^2 + 30x - 1.$$

Solución: Teniendo en cuenta que en este caso

$$1 + \frac{\text{máx}_{k=0,..,n-1} \mid a_k \mid}{\mid a_n \mid} = 1 + \frac{45}{20} = \frac{65}{20},$$

todas las raíces están en el intervalo $[-\frac{65}{20}, \frac{65}{20}]$. Para aislar las raíces calculamos los ceros de la derivada $P'(x) = 60x^2 - 90x + 30$, dichas raíces son 1 y 1/2. Por otro lado tenemos

$$P(-\frac{65}{20}) = -1260.4, \quad P(\frac{1}{2}) = \frac{21}{4}, \quad P(1) = 4, \quad P(\frac{65}{20}) = 307.75,$$

solo hay cambio de signo en el primer intervalo y por tanto hay una única raíz en el intervalo $[-\frac{65}{20}, \frac{1}{2}]$.

Problema 22 *Aislar en intervalos las raíces del polinomio* $P(x) = 2x^3+3x^2-12x+1$

Solución: teniendo en cuenta que en este caso

$$1 + \frac{\text{máx}_{k=0,..,n-1} \mid a_k \mid}{\mid a_n \mid} = 1 + \frac{12}{2} = 7,$$

todas las raíces están en el intervalo [-7,7]. Para aislar las raíces calculamos los ceros de la derivada $P'(x) = 6x^2+6x-12$, dichas raíces son 1 y -2. Por otro lado tenemos

$$P(-7) = -454, \quad P(-2) = 21, \quad P(1) = -6, \quad P(7) = 750,$$

hay cambio de signos en todos los intervalos y por tanto los intervalos donde están las raíces son $[-7,-2]$, $[-2,1]$ y $[1,7]$.

2.8. Aplicación en Epidemiología

En este apartado vamos a intentar responder a la pregunta : ¿Hasta cuantos días después de la infección por la COVID-19 pueden aparecer síntomas?. Para responder a esta pregunta necesitamos estimar la distribución de la probabilidad, $f(x)$, de que un paciente presente síntomas x días después de ser infectado. Para poder estimar esta probabilidad experimentalmente hay que hacer un estudio sobre una muestra de casos. Esto es lo que hicieron los investigadores Ma et al. (ver [Ma]) tomando una muestra de 587 casos de pacientes donde fue posible identificar el día que se infectaron y el día que presentaron síntomas. En las siguientes tablas se representa la muestra obtenida donde en la fila superior de cada tabla aparece, x, el número de días que han pasado desde la infección y en la fila inferior el número de pacientes que presentaron síntomas ese día. Así, por ejemplo, hay dos pacientes que presentaron síntomas el mismo día que se infectaron (lo cual es bastante excepcional), hay 76 pacientes que presentaron síntomas 4 días después de la infección y 1 paciente presentó síntomas 23 días después del contagio.

nº días después del contagio	0	1	2	3	4	5	6	7	8	9	10	11
nº de casos	2	26	31	43	76	47	47	54	47	46	38	27

nº días después del contagio	12	13	14	15	16	17	18	19	20	21	22	23
nº de casos	31	17	16	13	7	10	2	0	3	1	2	1

Una muestra es siempre algo imperfecto, pues es un número limitado de casos y además decidir el momento en que un paciente presenta síntomas puede incluir errores. Por ello, lo que se hace habitualmente es aproximar la muestra usando una distribución de probabilidad conocida que pueda adaptarse razonablemente bien a la muestra. Una distribución muy conocida, que además se usa mucho en Epidemiología, es la distribución Gamma, que se denota por $\Gamma(\alpha, \beta)$ y que depende de los parámetros α y β. La función de densidad de probabilidad, $f(x)$, de la distribución $\Gamma(\alpha, \beta)$ viene dada por

$$f(x) = \begin{cases} 0 & \text{si} \quad x \leq 0 \\ dx^{\alpha-1}e^{-\beta x} & \text{si} \quad x > 0 \end{cases} \tag{2.25}$$

donde d es una constante que depende de α y β y que se ajusta para que la suma de todas las probabilidades sea 1, lo cual nos da la condición:

$$d = \frac{1}{\int_0^\infty x^{\alpha-1}e^{-\beta x}dx}$$

La manera habitual de aproximar una muestra por una distribución consiste en ajustar los parámetros de la distribución para que tenga la misma media, m, y la misma varianza, V, que la muestra. La media y varianza muestrales en nuestro caso vienen dadas por

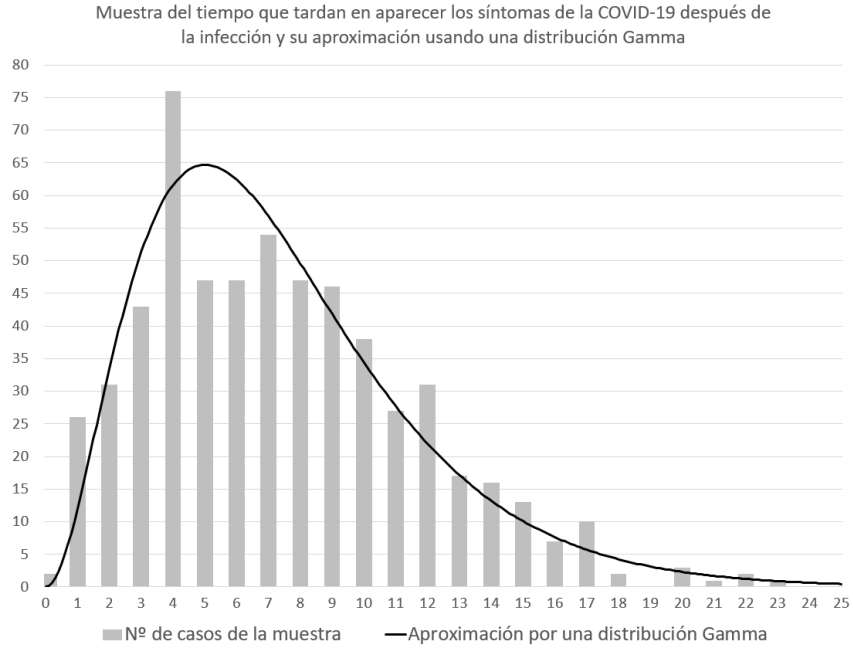
$$m = \frac{0 \cdot 2 + 1 \cdot 26 + 2 \cdot 31 + ... + 23 \cdot 1}{587} = 7.429302$$

$$V = \frac{(0-m)^2 \cdot 2 + (1-m)^2 \cdot 26 + (2-m)^2 \cdot 31 + ... + (23-m)^2 \cdot 1}{587} = 17.945172$$

la media y la varianza teórica de la distribución $\Gamma(\alpha, \beta)$ vienen dadas por $m = \alpha/\beta$ y $V = \alpha/\beta^2$. Igualando estas fórmulas a la media y varianza muestral calculadas podemos despejar fácilmente α y β obteniendo que $\alpha = 3.0757$ y $\beta = 0.414$. Además el valor de d en la fórmula (2.25) para estos valores de α y β viene dado por

$$d = \frac{1}{\int_0^\infty x^{2.0757} e^{-0.414x} dx} \approx 0.03091$$

En la gráfica siguiente se muestra, en columnas, los valores de la muestra utilizada y la distribución $\Gamma(\alpha, \beta)$ obtenida multiplicada por 587 para escalar la distribución al tamaño de la muestra. Como se observa en la gráfica la muestra es aproximada razonablemente bien por la distribución $\Gamma(\alpha, \beta)$.



Para responder a la pregunta que nos formulamos al principio de este apartado, es decir, hasta cuando se puede esperar que una persona presente síntomas, vamos a calcular cuando la probabilidad de presentar síntomas es muy pequeña, por ejemplo 0.01. Es decir, tenemos que calcular el valor de x para el cual

$$0.03091 x^{2.0757} e^{-0.414x} - 0.01 = 0$$

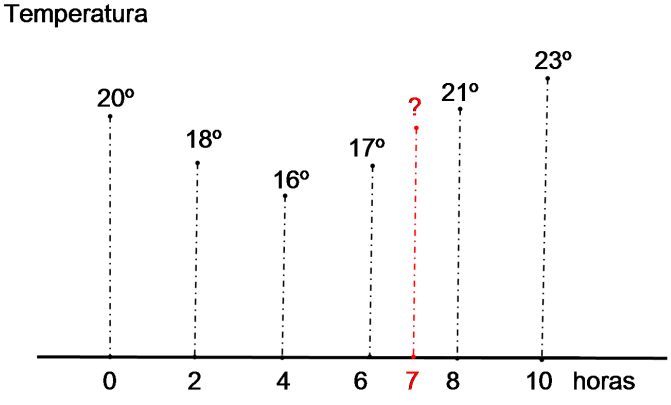
este valor, lo podemos calcular usando alguna de las técnicas que se han visto en este tema para el cálculo de ceros. Hay que tener en cuenta que esta ecuación va a tener 2 soluciones, una muy próxima a cero, que se produce cuando $f(x)$ empieza a subir y

otra, que es la que nos interesa, que es cuando $f(x)$ va bajando. Se puede comprobar que el resultado que se obtiene es apróximadamente $x = 16.902$, lo que significa que es muy poco probable que una persona presente síntomas pasados 16.902 días desde el momento de la infección.

Capítulo 3

INTERPOLACIÓN DE FUNCIONES I

El problema general de la interpolación de funciones consiste en, a partir del conocimiento del valor de una función en un conjunto finito de puntos, aproximar el valor de la función fuera de ese conjunto finito de puntos. Por ejemplo, en la siguiente figura se representan unos datos de temperatura tomados cada 2 horas. El problema de interpolación consiste en dar una estimación de la temperatura fuera de las horas en que se ha medido. Nótese que esta estimación, en general, no es exacta debido a que fuera de las horas tabuladas no se ha realizado una medición real, y por tanto el valor interpolado es una aproximación del valor real.

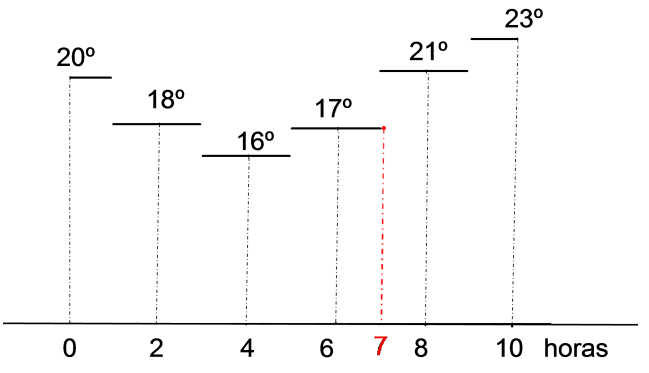


Los métodos de interpolación que veremos en este tema se dividen en dos grandes tipos: el primer tipo engloba los métodos en los que se divide la región de interés en intervalos y para cada intervalo se utiliza una fórmula diferente para interpolar. En el segundo tipo se calcula una única expresión, dada por un polinomio, que se ajusta a todos los puntos de interpolación. Si hay muchos puntos de interpolación se utiliza habitualmente las técnicas del primer tipo puesto que si intentamos ajustar un polinomio a todos los puntos nos daría un grado del polinomio alto que suelen

presentar fuertes oscilaciones. Además de estos dos tipos veremos la interpolación por mínimos cuadrados donde a la función interpolante no se le exige pasar por los puntos de interpolación.

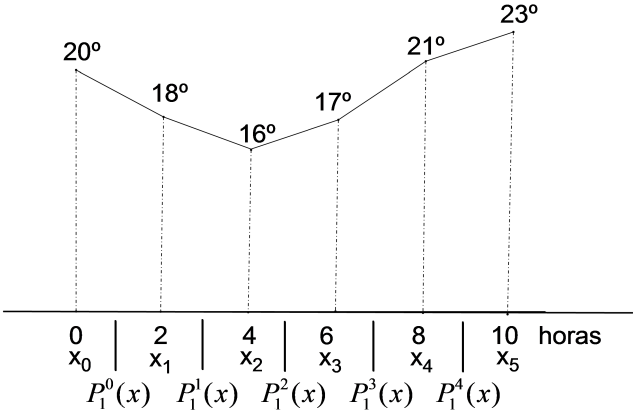
3.1. Interpolación de funciones por intervalos

Interpolación por el vecino más cercano



Este método es muy sencillo y consiste en calcular para cada $x \in R$ el punto de interpolación, x_i, más cercano a él y asignar como valor interpolado el valor de $f(x_i)$. Como puede verse en la gráfica anterior, este proceso genera una función interpolante discontinua en los puntos que están en la mitad de los intervalos dados por puntos de interpolación consecutivos. Además, en dichos puntos, la función no queda bien definida y su valor final depende de como se haya implementado el método.

Interpolación lineal (splines de grado 1)



$$P_1^i(x) = a_i + b_i(x - x_i)$$

Cada intervalo $[x_i, x_{i+1}]$ tiene asociado un polinomio de grado 1 de la forma

$$P_1^i(x) = a_i + b_i(x - x_i), \tag{3.1}$$

como la función que interpola tiene que pasar por los valores conocidos $f(x_i)$ y $f(x_{i+1})$ tiene que cumplirse

$$\begin{aligned} f(x_i) &= P_1^i(x_i) = a_i, \\ f(x_{i+1}) &= P_1^i(x_{i+1}) = a_i + b_i(x_{i+1} - x_i), \end{aligned} \tag{3.2}$$

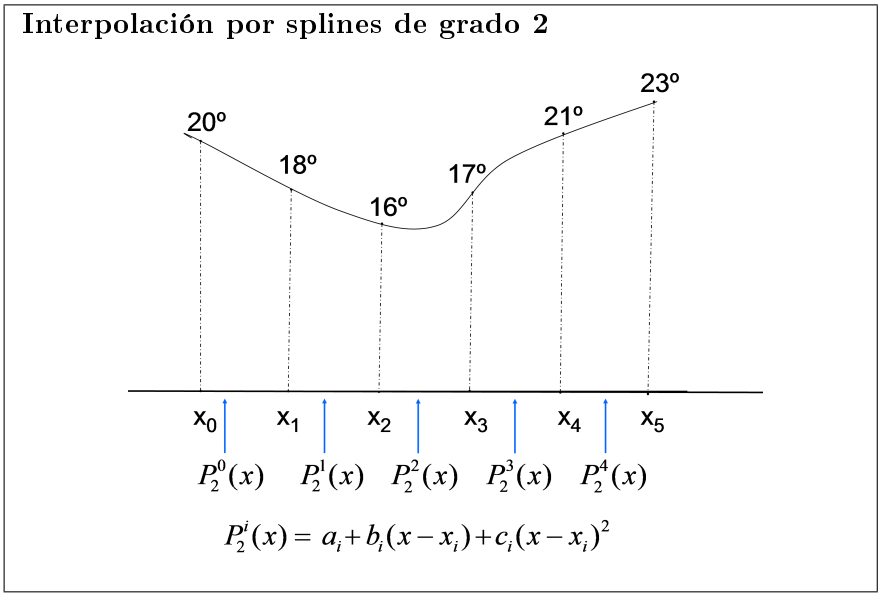
de donde despejando se obtiene

$$\begin{aligned} a_i &= f(x_i), \\ b_i &= \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}. \end{aligned} \tag{3.3}$$

Descripción algoritmo interpolación lineal.

- Dado un valor $x \in R$, se calcula el intervalo $[x_i, x_{i+1}]$ donde se encuentra x, si $x < x_0$ se toma el primer intervalo y si $x > x_n$ se toma el último intervalo.
- el valor interpolado en x es

$$P_1^i(x) = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i). \tag{3.4}$$



Cada intervalo $[x_i, x_{i+1}]$ tiene asociado un polinomio de grado 2 de la forma

$$P_2^i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2, \tag{3.5}$$

la función que interpola tiene que pasar por los valores conocidos $f(x_i)$ y $f(x_{i+1})$ y además imponemos que la derivada coincida a la izquierda y derecha de cada punto. Por tanto debe cumplirse

$$\begin{aligned} f(x_i) &= P_2^i(x_i) = a_i, \\ f(x_{i+1}) &= P_2^i(x_{i+1}) = a_i + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2, \\ \frac{d}{dx}P_2^i(x_{i+1}) &= \frac{d}{dx}P_2^{i+1}(x_{i+1}) \Rightarrow b_i + 2c_i(x_{i+1} - x_i) = b_{i+1}, \end{aligned} \tag{3.6}$$

de la primera ecuación salen directamente los valores de a_i, para calcular b_i y c_i por cada intervalo $[x_i, x_{i+1}]$ tenemos 2 ecuaciones (la segunda y tercera ecuación), salvo para el último intervalo que la tercera ecuación no está al no existir un intervalo por la derecha con el que conectar la derivada. Por tanto el sistema está incompleto y hay que añadir alguna información adicional para poder calcular la solución. Como veremos a continuación, una vez que para cualquier intervalo queda fijado el polinomio, automáticamente se pueden calcular todos los demás. Efectivamente si conocemos b_i y c_i podemos calcular b_{i+1} y c_{i+1} teniendo en cuenta que b_{i+1} sale de la tercera ecuación de (3.6) y c_{i+1} sale despejando de

$$a_{i+1} + b_{i+1}(x_{i+2} - x_{i+1}) + c_{i+1}(x_{i+2} - x_{i+1})^2 = f(x_{i+2}),$$

de donde sale

$$c_{i+1} = \frac{f(x_{i+2}) - f(x_{i+1}) - b_{i+1}(x_{i+2} - x_{i+1})}{(x_{i+2} - x_{i+1})^2}. \tag{3.7}$$

Por otro lado, si conocemos b_i y c_i podemos calcular b_{i-1} y c_{i-1} resolviendo el sistema

$$\begin{aligned} a_{i-1} + b_{i-1}(x_i - x_{i-1}) + c_{i-1}(x_i - x_{i-1})^2 &= f(x_i), \\ b_{i-1} + 2c_{i-1}(x_i - x_{i-1}) &= b_i, \end{aligned} \tag{3.8}$$

de donde b_{i-1}, c_{i-1} se pueden despejar y se obtiene

$$\begin{aligned} b_{i-1} &= \frac{2\left(f(x_i) - f(x_{i-1})\right)}{x_i - x_{i-1}} - b_i, \\ c_{i-1} &= \frac{b_i(x_i - x_{i-1}) + f(x_{i-1}) - f(x_i)}{(x_i - x_{i-1})^2}. \end{aligned} \tag{3.9}$$

Por último, si conocemos uno de los valores b_i ó c_i el otro valor se puede obtener despejando de la segunda ecuación de (3.6). Por tanto fijando como parámetro un b_i ó c_i cualquiera, todos los demás quedan determinados. Por ejemplo fijando $c_0 = 0$ todos los valores de b_i y c_i quedan determinados.

Descripción algoritmo interpolación por splines de grado 2 (c_0 conocido).

- Se asigna $a_i = f(x_i)$ y se calcula b_0 despejando de la segunda ecuación de (3.6) obteniendo

$$b_0 = \frac{f(x_1) - f(x_0) - c_0(x_1 - x_0)^2}{x_1 - x_0}. \tag{3.10}$$

- Se obtiene el resto de valores b_i, c_i de forma iterativa usando las fórmulas

$$\begin{aligned} b_{i+1} &= b_i + 2c_i(x_{i+1} - x_i), \\ c_{i+1} &= \frac{f(x_{i+2}) - f(x_{i+1}) - b_{i+1}(x_{i+2} - x_{i+1})}{(x_{i+2} - x_{i+1})^2}. \end{aligned} \tag{3.11}$$

- Dado un valor $x \in R$, se calcula el intervalo $[x_i, x_{i+1}]$ donde se encuentra x, si $x < x_0$ se toma el primer intervalo y si $x > x_n$ se toma el último intervalo. El valor interpolado en x es

$$P_2^i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2. \tag{3.12}$$

Descripción algoritmo interpolación por splines de grado 2 (c_{n-1} conocido).

- Se asigna $a_i = f(x_i)$ y se calcula b_{n-1} despejando de la segunda ecuación de (3.6) obteniendo

$$b_{n-1} = \frac{f(x_n) - f(x_{n-1}) - c_{n-1}(x_n - x_{n-1})^2}{x_n - x_{n-1}}. \tag{3.13}$$

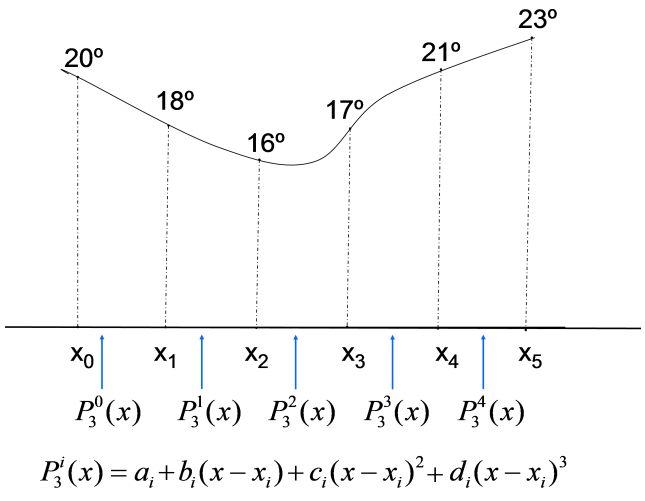
- Se obtiene el resto de valores b_i, c_i de forma iterativa usando las fórmulas

$$\begin{aligned} b_{i-1} &= \frac{2\left(f(x_i) - f(x_{i-1})\right)}{x_i - x_{i-1}} - b_i, \\ c_{i-1} &= \frac{b_i(x_i - x_{i-1}) + f(x_{i-1}) - f(x_i)}{(x_i - x_{i-1})^2}. \end{aligned} \tag{3.14}$$

- Dado un valor $x \in R$, se calcula el intervalo $[x_i, x_{i+1}]$ donde se encuentra x, si $x < x_0$ se toma el primer intervalo y si $x > x_n$ se toma el último intervalo. El valor interpolado en x es

$$P_2^i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2. \tag{3.15}$$

Interpolación por splines cúbicos.



Se define un polinomio de grado 3 distinto $P_3^i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x-x_i)^3$ para cada intervalo $[x_i, x_{i+1}]$. Si hay $N+1$ puntos, el número de polinomios es N. Para definir estos polinomios, se imponen las siguientes condiciones:

$$\begin{aligned} P_3^i(x_i) &= f(x_i) \quad i = 0, .., N-1, \\ P_3^i(x_{i+1}) &= f(x_{i+1}) \quad i = 0, ..., N-1, \\ \frac{\partial P_3^i}{\partial x}(x_{i+1}) &= \frac{\partial P_3^{i+1}}{\partial x}(x_{i+1}) \quad i = 0, .., N-2, \\ \frac{\partial^2 P_3^i}{\partial x^2}(x_{i+1}) &= \frac{\partial^2 P_3^{i+1}}{\partial x^2}(x_{i+1}) \quad i = 0, ..., N-2. \end{aligned} \tag{3.16}$$

Vamos a introducir la notación $h_i = x_{i+1} - x_i$. Nótese que, para definir los polinomios, tenemos que buscar $4N$ valores, es decir: $a_0, ..., a_{N-1}, b_0, ..., b_{N-1}, c_0, ..., c_{N-1}$, $d_0, ..., d_{N-1}$. Por razones técnicas al escribir las fórmulas, vamos a utilizar también los valores a_N y c_N (que serían coeficientes del intervalo siguiente a x_N (que no existe)). $a_N = f(x_N)$ y c_N se define para que cumpla la condición de la segunda derivada si existiera un intervalo a la derecha de x_N, es decir:

$$c_N = \frac{1}{2}\frac{\partial^2 P_3^{N-1}}{\partial x^2}(x_N). \tag{3.17}$$

Teorema 8 *Si una familia de polinomios* $P_3^i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$, $i = 0, .., N-1$, *satisface las condiciones anteriores, entonces*

$$a_i = f(x_i) \quad i = 0, .., N, \tag{3.18}$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i} \quad i = 0, .., N-1, \tag{3.19}$$

$$b_i = \frac{f_{i+1} - f_i}{h_i} - \frac{h_i\,(2c_i + c_{i+1})}{3} \quad i = 0, .., N-1, \tag{3.20}$$

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = \frac{3(f_{i+1} - f_i)}{h_i} - \frac{3\,(f_i - f_{i-1})}{h_{i-1}}, \tag{3.21}$$

para $i = 1, .., N-1$.

Demostración: De la condición $P_3^i(x_i) = f(x_i)$, se obtiene de forma inmediata que $a_i = f(x_i)$. De la condición $\frac{\partial^2 P_3^{i+1}}{\partial x^2}(x_{i+1}) = \frac{\partial^2 P_3^i}{\partial x^2}(x_{i+1})$, se obtiene que

$$2c_{i+1} = 6d_i h_i + 2c_i \tag{3.22}$$

de donde, despejando, obtenemos que

$$d_i = \frac{c_{i+1} - c_i}{3h_i}. \tag{3.23}$$

De la condición $P_3^i(x_{i+1}) = f(x_{i+1})$, se obtiene que

$$d_i h_i^3 + c_i h_i^2 + b_i h_i + f_i = f_{i+1}. \tag{3.24}$$

Despejando, obtenemos que

$$b_i = \frac{a_{i+1} - a_i}{h_i} - d_i h_i^2 - c_i h_i = \frac{f_{i+1} - f_i}{h_i} - \frac{h_i\,(2c_i + c_{i+1})}{3}. \tag{3.25}$$

Finalmente, de la condición $\frac{\partial P_3^i}{\partial x}(x_i) = \frac{\partial P_3^{i-1}}{\partial x}(x_i)$, se obtiene que

$$b_i = 3d_{i-1}h_{i-1}^2 + 2c_{i-1}h_{i-1} + b_{i-1}, \tag{3.26}$$

y, despejando todo en función de c_i, se obtiene la relación

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = \frac{3(f_{i+1} - f_i)}{h_i} - \frac{3\,(f_i - f_{i-1})}{h_{i-1}}. \tag{3.27}$$

Nótese que esta última relación determina un sistema de ecuaciones donde las incógnitas son las variables c_i. Dicho sistema tiene $N+1$ incógnitas $(c_0, ..., c_N)$ y $N-1$ ecuaciones. Para completar dicho sistema hay que añadir dos ecuaciones, normalmente se añade una ecuación que involucre a c_0 y otra ecuación que involucre a c_N. Para añadir estas dos ecuaciones se puede simplemente fijar como parámetros de la función los valores c_0 y c_N. Esto se puede añadir al sistema de ecuaciones anterior añadiendo al principio la ecuación $c_0 = c0$ y al final la ecuación $c_N = cN$, donde $c0$ y cN son los valores preasignados como parámetros. Por tanto el algoritmo de interpolación de splines de grado 3 se puede describir de la siguientes forma :

Descripción algoritmo interpolación por splines de grado 3 ($c_0 = c0$ y $c_N = cN$).

- se calcula el vector $h_i = x_{i+1} - x_i$
- se plantea un sistema lineal tridiagonal para calcular los c_i. Para ello tenemos en cuenta que :
  - la diagonal principal del sistema que llamaremos M_i vale $M_0 = M_N = 1$ y en el resto de los casos
  $$M_i = 2(h_{i-1} + h_i)$$
  - la diagonal inferior del sistema que llamaremos L_i vale $L_{N-1} = 0$ y para $i < N - 1$
  $$L_i = h_i$$
  - la diagonal superior del sistema que llamaremos U_i vale $U_0 = 0$ y para $i > 0$
  $$U_i = h_i$$
  - el término independiente del sistema, que llamaremos B_i vale $B_0 = c0$, $B_N = cN$, y en el resto de los casos
  $$B_i = \frac{3(f_{i+1} - f_i)}{h_i} - \frac{3\,(f_i - f_{i-1})}{h_{i-1}}.$$
- se calcula c_i resolviendo el sistema resultante usando una función previamente implementada a la que pasamos como parámetros los vectores M, L, U y B.
- una vez obtenido el vector c_i se calculan a_i, b_i y d_i usando las ecuaciones (3.20).
- Dado un valor $x \in R$, se calcula el intervalo $[x_i, x_{i+1}]$ donde se encuentra x, si $x < x_0$ se toma el primer intervalo y si $x > x_n$ se toma el último intervalo. El valor interpolado en x es
$$P_3^i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3. \tag{3.28}$$

Otra opción habitual es fijar los valores de $f'(a)$ y $f'(b)$ y extraer las ecuaciones correspondientes para completar el sistema de ecuaciones. Por lo tanto, siguiendo con el resultado del teorema anterior, para calcular los splines cúbicos es necesario, en primer lugar, tomar $a_i = f(x_i)$. A continuación, se resuelve un sistema de ecuaciones tridiagonal para el cálculo de los c_i. Finalmente, los b_j y d_j se calculan directamente a partir de las relaciones mostradas en el teorema anterior.

Ejemplo 3 *Vamos a calcular los polinomios interpoladores utilizando splines cúbicos al interpolar la función $f(x)$ en los puntos $x = 0, 1, 2$ y 3, sabiendo que $f(0) = 0$, $f(1) = 1$, $f(2) = 0$ y $f(3) = 2$, tomando $c_0 = c_3 = 0$. En este caso $h_i = 1$. Debemos definir 3 polinomios distintos que corresponden a los intervalos $[0, 1]$, $[1, 2]$, y $[2, 3]$. Los términos a_i vienen dados por*

$$a_0 = 0, \quad a_1 = 1, \quad a_2 = 0, \quad a_3 = 2. \tag{3.29}$$

El sistema que debemos resolver para calcular los c_i es

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -6 \\ 9 \\ 0 \end{pmatrix} \quad \text{cuya solución es} \quad \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -2.2 \\ 2.8 \\ 0 \end{pmatrix}. \tag{3.30}$$

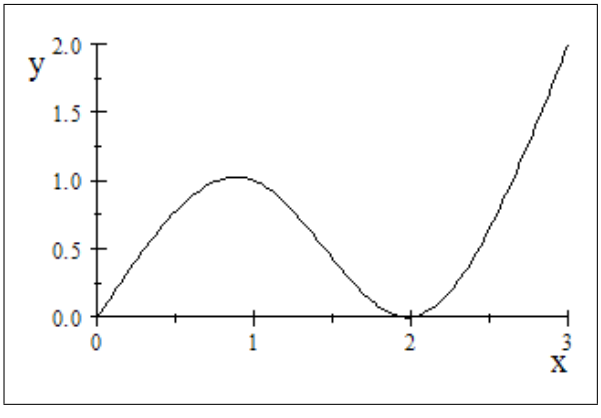
Los valores b_i y d_i vienen dados por

$$\begin{array}{ll} b_0 = 1 - \frac{-2.2}{3} = 1.733, & d_0 = \frac{-2.2-0}{3} = -0.733, \\ b_1 = -1 - \frac{-4.4+2.8}{3} = -0.467, & d_1 = \frac{2.8+2.2}{3} = 1.667, \\ b_2 = 2 - \frac{5.6+0}{3} = 0.133, & d_2 = \frac{0.-2.8}{3} = -0.933, \end{array} \tag{3.31}$$

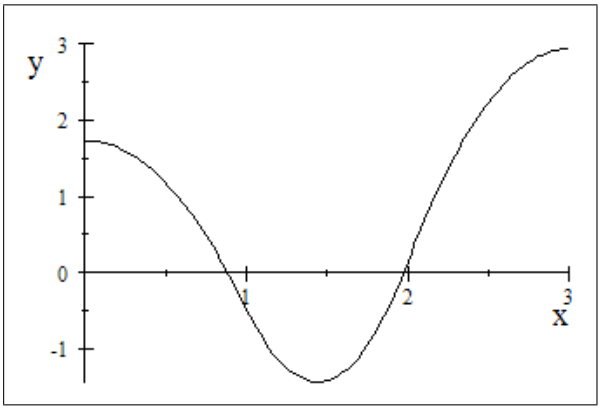
por tanto, los polinomios son

$$\begin{array}{lll} P_0(x) = -0.733x^3 + 1.733x & \text{si} & x \in [0, 1], \\ P_1(x) = 1.667\,(x-1)^3 - 2.2\,(x-1)^2 - 0.467\,(x-1) + 1 & \text{si} & x \in [1, 2], \\ P_2(x) = -0.933\,(x-2)^3 + 2.8\,(x-2)^2 + 0.133\,(x-2) & \text{si} & x \in [2, 3]. \end{array} \tag{3.32}$$

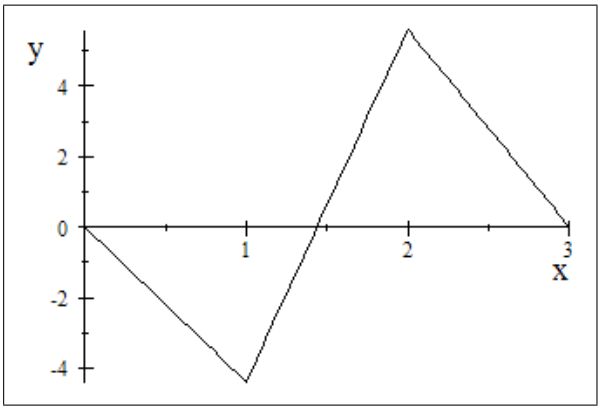
A continuación se muestra una gráfica con los 3 polinomios concatenados en el intervalo $[0, 3]$:



Como puede observarse, por las condiciones sobre las derivadas que hemos impuesto, no es posible distinguir geométricamente, al trazar la curva, cuales son los puntos de unión entre los tres polinomios. Es decir, parece, a simple vista, el trazado de una única función. Veamos ahora gráficamente el perfil de la derivada de los polinomios $P_0(x)$, $P_1(x)$, y $P_2(x)$.



como puede observarse, tampoco sobre la derivada se aprecian los puntos de unión de los polinomios. Sin embargo, sobre la gráfica de la derivada segunda los puntos de unión se detectan en los lugares donde encontramos un pico, tal y como se muestra en la gráfica de la derivada segunda siguiente:



3.2. Aplicación de la interpolación por splines a la interpolación de curvas 2D

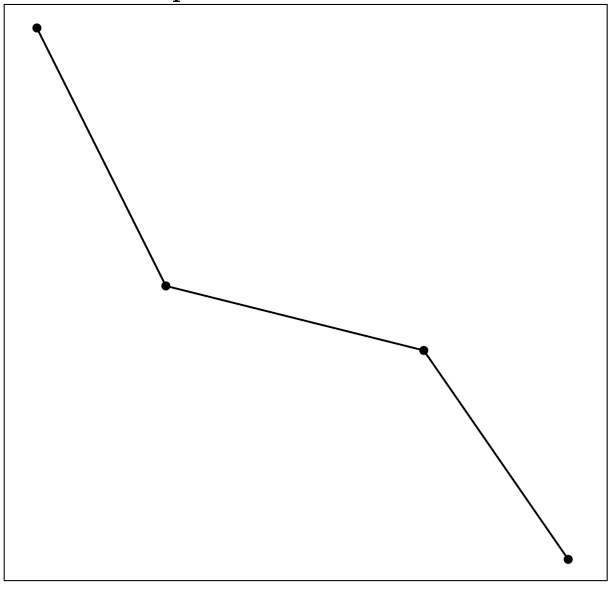
Una curva en 2D en forma paramétrica viene dada por una función continua $C(t) = (x(t), y(t))$ que determina el trazo de la curva al dibujarla en el plano. Por ejemplo, una circunferencia viene dada por $C(t) = (Rcos(t), Rsin(t))$. Las curvas 2D son de gran importancia en gráficos por ordenador porque los contornos de los objetos gráficos que se visualizan en la pantalla del ordenador vienen dados por curvas 2D que conectan unas con otras. Por ejemplo, el contorno del logo de la ULPGC, basado en la obra de "El pensador" del escultor Martín Chirino es la combinación de varias curvas. El proceso de creación de estas curvas, por parte de los diseñadores gráficos consiste, habitualmente, en fijar unos puntos de referencia y posteriormente definir unas curvas que unen esos puntos usando las herramientas que suministran las aplicaciones de diseño gráfico. A continuación se muestra el logo de la ULPGC

donde se aprecia claramente este proceso de combinación de curvas para crear el logo.

Logo de la ULPGC

En esta sección vamos a aprender como podemos utilizar las técnicas de interpolación por splines para interpolar curvas 2D. Nuestro punto de partida son algunos puntos (x_k, y_k) situados sobre la curva que queremos interpolar. Por ejemplo, en la figura siguiente se ilustran 4 puntos de referencia que vamos a usar que inicialmente hemos unido por segmentos.

4 puntos de referencia para realizar la interpolación de una curva 2D



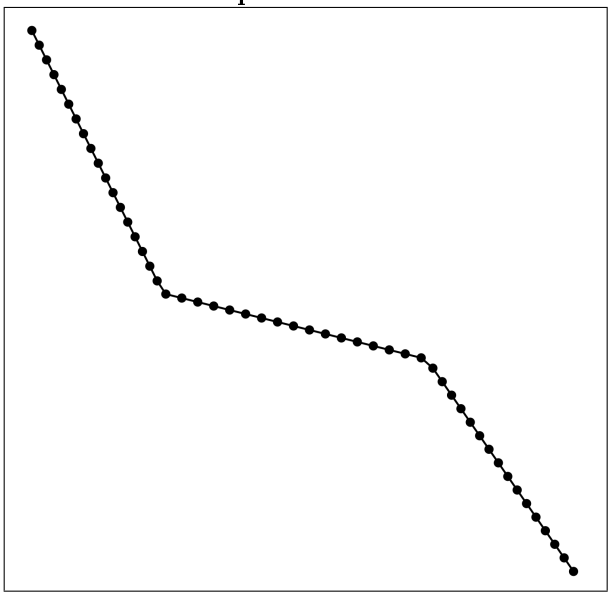
Nótese que la variable t no está definida, es decir, normalmente no nos dicen para que valor de t la curva va a pasar por esos puntos. Esto es así porque normalmente el

valor de t no es muy relevante, lo importante es el trazo de la curva. Es decir, el trazo de la curva $(Rcos(t), Rsin(t))$ es el mismo que el de la curva $(Rcos(2t^2), Rsin(2t^2))$. Normalmente, lo que se hace es asociar, para cada punto (x_k, y_k) un valor t_k en función de la longitud de los segmentos. Es decir, se toma $t_0 = 0$ y para $k > 0$ definimos

$$t_k = t_{k-1} + \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}$$

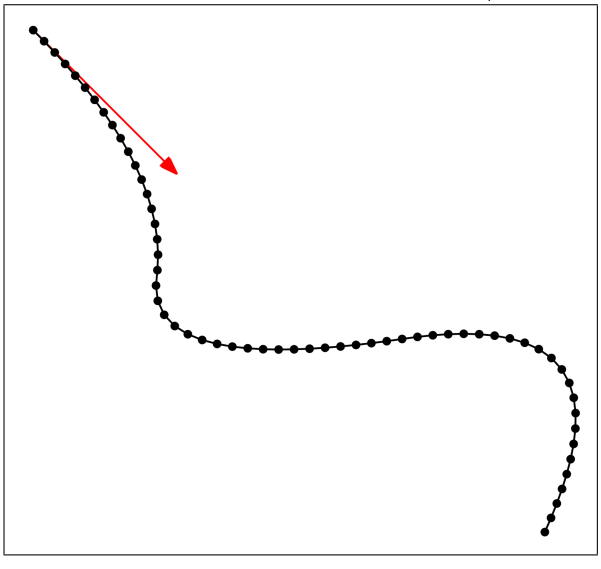
Para interpolar la curva entre esos puntos utilizaremos las técnicas por splines vistas aplicadas por separado a las secuencias (t_k, x_k) y (t_k, y_k). Para visualizar el resultado vamos a dibujar puntos apróximadamente equidistantes situados sobre la curva generada. Por ejemplo, en la siguiente figura se muestra el resultado usando la interpolación lineal, como vemos, los puntos nuevos generados se encuentran sobre los segmentos que unen los puntos originales.

Interpolación lineal



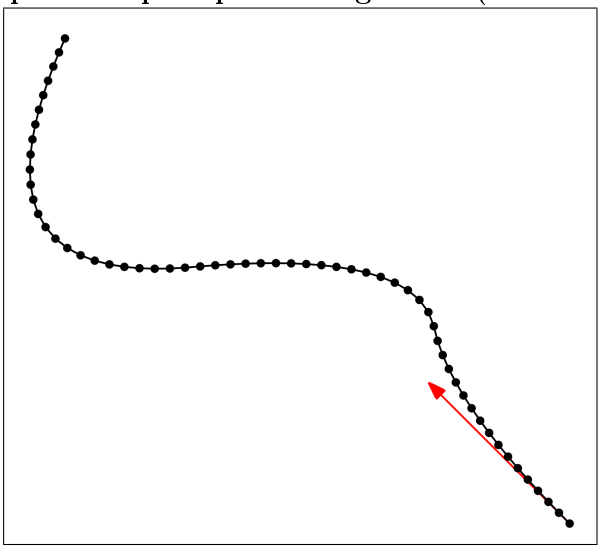
En la siguiente figura se muestra el resultado usando la interpolación por splines de grado 2 con control del punto inicial. Se dibuja la tangente a la curva en el punto inicial para indicar que podemos controlar el resultado del trazo de la curva cambiando dicho vector tangente. Un diseñador gráfico movería ese vector tangente hasta obtener el resultado deseado.

Interpolación por splines de grado 2 (variante c_0)



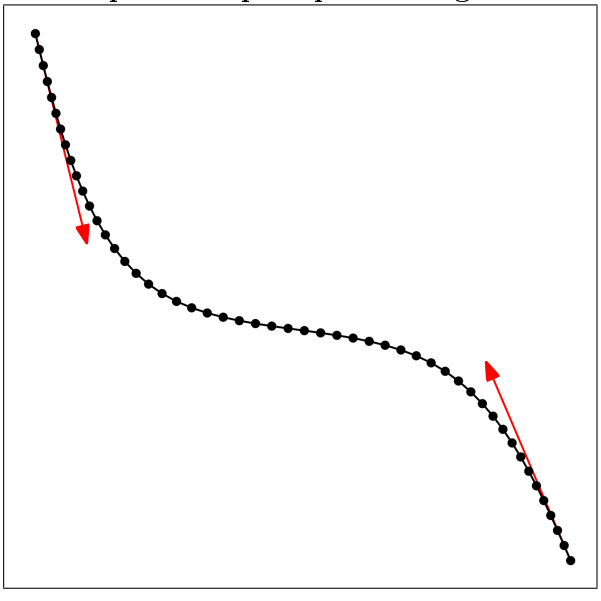
En la siguiente figura se muestra lo mismo pero controlando el vector tangente en el último punto

Interpolación por splines de grado 2 (variante c_N)



Por último, en la siguiente figura se muestra el resultado con los splines de grado 3. Se dibujan vectores tangentes al principio y al final de la curva porque los splines de grado 3 nos permiten controlar a la vez como sale la curva al principio y como termina.

Interpolación por splines de grado 3



Observamos que los splines de grado 3 suministran el mejor resultado porque nos permiten tener un mayor control sobre el resultado final del trazo de la curva. De hecho, estos splines de grado 3, son en la práctica, una técnica muy utilizada en diseño gráfico para interpolar curvas, aunque se expresan en otro formato que

se denomina curvas cúbicas de Bézier, pero que teóricamente son equivalentes a los splines de grado 3.

En las prácticas, para visualizar estas curvas utilizaremos el formato SVG, que es un formato estándar en diseño gráfico que tiene la ventaja que se puede visualizar usando cualquier navegador. Aunque nosotros no explotamos ese aspecto aquí, el formato SVG es un formato de gráficos vectorial. Es decir, no es necesario suministrarles las curvas discretizadas a través de una colección de puntos situados sobre la curva (como hemos hecho nosotros en las figuras anteriores). En un formato vectorial podemos darle los parámetros (por ejemplo los coeficientes del polinomio) que determinan el trazo continuo de la curva y el software de renderizado realiza el trazo de la curva de acuerdo con la precisión adecuada en cada caso. Por ejemplo, si visualizamos el logo de la ULPGC en un navegador en un formato vectorial, podemos hacer un zoom en el navegador tan grande como queramos y la curvas del contorno del logo siempre se verán perfectas. Si utilizaramos un formato de puntos discretos para representar el logo (por ejemplo si lo tenemos guardado como una imagen) al hacer un zoom grande, empezaríamos a ver los contornos del logo pixelados.

3.3. Interpolación por polinomios de Lagrange.

Sea una función $f(x)$ de la que conocemos sus valores en un conjunto finito de puntos $\{x_i\}_{i=0,..,N}$. Es decir, sabemos que $f(x_i) = f_i$. El polinomio interpolador de Lagrange $P_N(x)$ de $f(x)$ en los puntos $\{x_i\}_{i=0,..,N}$ es el único polinomio de grado menor o igual que N tal que

$$P_N(x_i) = f(x_i) \quad \forall i = 0, .., N. \tag{3.33}$$

$P_N(x)$ se puede expresar en término de los denominados polinomios base de Lagrange $P^i(x)$, definidos como:

$$P^i(x) = \frac{\Pi_{j\neq i}^N (x - x_j)}{\Pi_{j\neq i}^N (x_i - x_j)}, \tag{3.34}$$

estos polinomios base tienen la propiedad fundamental siguiente

$$P^i(x_j) = \begin{cases} 1 & si \quad i = j, \\ 0 & si \quad i \neq j. \end{cases} \tag{3.35}$$

Por tanto, el polinomio interpolador de Lagrange puede expresarse como

$$P_N(x) = \sum_{i=0}^{N} f(x_i) P^i(x). \tag{3.36}$$

Ejemplo 4 *Consideremos la función $f(x) = e^x$, vamos a interpolarla en los puntos $x_0 = 0$, $x_1 = -1$ y $x_2 = 1$. Para calcular $P_2(x)$, el polinomio interpolador de Lagrange en estos puntos, calculamos los polinomios base:*

$$P^0(x) = \frac{(x+1)(x-1)}{-1}, \qquad P^1(x) = \frac{x(x-1)}{2}, \qquad P^2(x) = \frac{x(x+1)}{2}, \tag{3.37}$$

siendo el polinomio interpolador:

$$P_2(x) = e^0 \frac{(x+1)(x-1)}{-1} + e^{-1}\frac{x(x-1)}{2} + e\frac{x(x+1)}{2}. \tag{3.38}$$

Teorema 9 *El polinomio interpolador de Lagrange es el único polinomio de grado igual o inferior a N tal que*

$$P_N(x_i) = f(x_i) \quad \forall i = 0, .., N, \tag{3.39}$$

Demostración Sea $P(x)$ un polinomio de grado inferior o igual a N que verifique que $P(x_i) = f(x_i) \quad \forall i = 0, .., N$. Entonces, el polinomio $Q(x) = P(x) - P_N(x)$ es un polinomio de grado inferior o igual a N que verifica que $Q(x_i) = 0$ y, por tanto, posee $N+1$ raíces, lo cual es imposible, salvo que $Q(x)$ sea idénticamente igual a cero. Por tanto $Q(x) \equiv 0$ y $P(x) = P_N(x)$.

3.3.1. Error de interpolación de Lagrange y polinomios de Chebychev.

Al aproximar $f(x)$ por el polinomio interpolador $P_N(x)$ en un intervalo $[a, b]$ se comete un error de interpolación, que viene determinado por el siguiente teorema.

Teorema 10 *Sea $f(x)$ una función, y $P_N(x)$ su polinomio interpolador de Lagrange en los puntos $\{x_i\}_{i=0,..,N} \subset [a, b]$ y $x \in [a, b]$, entonces*

$$f(x) - P_N(x) = \frac{f^{N+1)}(\xi)}{(N+1)!}\Pi_{i=0}^{N}(x - x_i), \tag{3.40}$$

donde ξ es un valor intermedio perteneciente a $[a, b]$.

Demostración: Si $x = x_i$, el error de interpolación es cero y por tanto la fórmula anterior es válida. Consideremos ahora x distinto a los x_i y definamos

$$\begin{aligned} w(t) &= \Pi_{i=0}^{N}(t - x_i), \\ \lambda &= \frac{f(x) - P_N(x)}{w(x)}, \\ \phi(t) &= f(t) - P_N(t) - \lambda w(t). \end{aligned} \tag{3.41}$$

La función $\phi(t)$ tiene al menos $N+1$ ceros en los puntos x_i y en el punto x. Por tanto, su función derivada $\phi'(t)$ tiene al menos N ceros repartidos entre los ceros de $\phi(t)$. Análogamente, $\phi''(t)$ tiene al menos $N-1$ ceros y así sucesivamente hasta llegar a $\phi^{N+1}(t)$, que tiene al menos 1 cero. Si llamamos ξ a dicho cero, obtenemos

$$\phi^{N+1}(\xi) = f^{N+1)}(\xi) - \lambda(N+1)!, \tag{3.42}$$

de donde, despejando y sustituyendo λ por su valor, obtenemos el resultado del Teorema. A continuación veremos una acotación del error de interpolación cuando los puntos de interpolación son equidistantes

Teorema 11 *Si los puntos* $\{x_i\}_{i=0,..,N}$ *son equidistantes, es decir*

$$x_{i+1} = x_i + h \quad donde \quad h = \frac{b-a}{N},$$

entonces :

$$max_{x\in[a,b]}|f(x) - P_N(x)| \leq \frac{max_{\xi\in[a,b]}|f^{N+1)}(\xi)|}{4(N+1)} h^{N+1}.$$

Demostración: Vamos a acotar $\left|\Pi_{i=0}^{N}(x-x_i)\right|$, se puede demostrar que esta función alcanza el máximo en el primer intervalo $[x_0, x_1]$. Además si $x \in [x_0, x_1]$

$$\left|\Pi_{i=0}^{N}(x-x_i)\right| \leq |(x-x_0)(x-x_0-h)|\, 2h \cdot 3h \cdots Nh,$$

además, la parábola $(x-x_0)(x-x_0-h)$ alcanza su extremo en $x_0 + \frac{h}{2}$ y por tanto tenemos

$$max_{x\in[a,b]}|f(x)-P_N(x)| \leq \frac{max_{\xi\in[a,b]}|f^{N+1)}(\xi)|}{(N+1)!} h^{N+1} N! \frac{}{4} = \frac{max_{\xi\in[a,b]}|f^{N+1)}(\xi)|}{4(N+1)} h^{N+1}.$$

La cuestión que vamos a abordar en el siguiente apartado es, en el caso en que queramos interpolar una función en un intervalo $[a, b]$, y que nosotros podamos elegir los valores de interpolación x_i, cómo elegirlos de tal forma que el error de interpolación sea mínimo. Para ello, elegiremos los puntos x_i tales que $max_{x\in[a,b]} \mid \Pi_{i=0}^{N}(x-x_i) \mid$ sea lo más pequeño posible en $[a, b]$, de acuerdo con el siguiente teorema:

Teorema 12 *Sea* $N \geq 0$, *y un intervalo* $[a, b]$ *Se consideran los puntos* x_i *dados por*

$$x_i = a + \frac{b-a}{2}\left(1 + \cos\left(\frac{2i+1}{2N+2}\pi\right)\right) \quad i = 0, ..., N, \tag{3.43}$$

entonces

$$max_{x\in[a,b]} \mid \Pi_{i=0}^{N}(x-x_i) \mid = \left(\frac{b-a}{2}\right)^{N+1} \frac{1}{2^N} \leq max_{x\in[a,b]} \mid \Pi_{j=0}^{N}(x-\widetilde{x}_j) \mid, \tag{3.44}$$

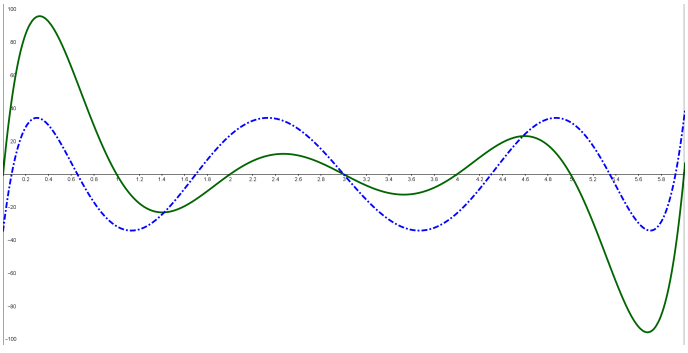
para cualquier otra elección posible de valores de interpolación $\widetilde{x}_j$.

Demostración: la demostración para el intervalo $[-1, 1]$ se encuentra en [Ki-Ch]. La demostración para un intervalo cualquiera $[a, b]$ se obtiene fácilmente transformando el intervalo $[-1, 1]$ en $[a, b]$.

Por tanto, utilizando este resultado, el error de interpolación máximo viene determinado por:

$$\mid f(x) - P_N(x) \mid \leq \frac{max_{x\in[a,b]} f^{N+1)}(\xi)}{(N+1)!2^N} \left(\frac{b-a}{2}\right)^{N+1}. \tag{3.45}$$

En la siguiente gráfica se ilustra la diferencia entre tomar los puntos equidistantes y elegirlos de forma óptima de acuerdo con el teorema anterior:



Para $N = 6$, se presentan las funciones $E(x) = \Pi_{i=0}^{6}(x - i)$ tomando puntos equidistantes en el intervalo [0,6] (trazo continuo), y la función $\hat{E}(x) = \Pi_{i=0}^{6}(x - x_i)$ tomando los puntos de interpolación óptimos x_i en [0,6] dados por la expresión (3.43) (trazo discontinuo). Tal y como se mencionó anteriormente, en el caso de puntos equidistantes el máximo de la función se alcanza en el primer intervalo. También se observa como al tomar los puntos óptimos el polinomio resultante es menos oscilante y su amplitud máxima es menor.

Ejemplo 5 *Se considera* $[a, b] = [0, 1]$ *y* $N = 5$ *(es decir* 6 *puntos de interpolación). Los puntos de interpolación dados por el teorema anterior son:*

$$\begin{array}{lll} x_0 = 0.98296, & x_1 = 0.85355, & x_2 = 0.62941, \\ x_3 = 0.37059, & x_4 = 0.14645, & x_5 = 0.017037 \end{array} \tag{3.46}$$

En el caso de que $[a, b] = [-1, 1]$, los valores óptimos de interpolación x_i dados por la fórmula anterior son las raíces de los denominados polinomios de Chebychev, $T_N(x)$, construidos recursivamente de la manera siguiente:

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_N(x) &= 2xT_{N-1}(x) - T_{N-2}(x). \end{aligned} \tag{3.47}$$

3.3.2. Método de diferencias de Newton para calcular el polinomio interpolador de Lagrange.

Numéricamente, el cálculo de $P_N(x)$ a través de los polinomios base necesita de la evaluación de $N+1$ polinomios de grado N. Además, si queremos añadir un nuevo punto de interpolación, debemos cambiar todos los polinomios base de Lagrange. Un método más directo para el cálculo de $P_N(x)$ es el denominado método de diferencias de Newton. El método consiste en ir calculando progresivamente los polinomios $P_k(x)$

que interpolan la función en los puntos $x_0, ..., x_k$ de la siguiente forma:

$$\begin{aligned} P_0(x) &= a_0, \\ P_1(x) &= P_0(x) + a_1(x - x_0), \\ P_2(x) &= P_1(x) + a_2(x - x_0)(x - x_1), \\ &... \\ P_N(x) &= P_{N-1}(x) + a_N(x - x_0)(x - x_1)...(x - x_{N-1}). \end{aligned} \tag{3.48}$$

A los coeficientes a_k los denotamos por

$$a_k = f[x_0, ..., x_k]. \tag{3.49}$$

Ejemplo 6 *Vamos a interpolar la función* $f(x) = e^x$ *en los puntos* $x_0 = 0$, $x_1 = 1$ *y* $x_2 = 2$.

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= 1 + a_1 x. \end{aligned} \tag{3.50}$$

Como $P_1(1)$ debe ser igual a e, despejando obtenemos

$$a_1 = e - 1.$$

Por último

$$P_2(x) = P_1(x) + a_2 x(x - 1). \tag{3.51}$$

Como $P_2(2)$ debe ser igual a e^2, despejando obtenemos

$$a_2 = \frac{e^2 - P_1(2)}{2}. \tag{3.52}$$

Por tanto, el polinomio $P_2(x)$ lo expresamos como

$$P_2(x) = 1 + (e - 1)x + \frac{e^2 - 2e + 1}{2} x(x - 1). \tag{3.53}$$

Como veremos en el teorema siguiente, los coeficientes $f[x_0, ..., x_k]$, que se denominan diferencias divididas de Newton, verifican las siguientes propiedades:

$$\begin{aligned} f[x_i] &= f(x_i), \\ f[x_i, x_{i+1}] &= \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}, \\ &. \\ f[x_i, .., x_{i+k}] &= \frac{f[x_{i+1}, .., x_{i+k}] - f[x_i, .., x_{i+k-1}]}{x_{i+k} - x_i}. \end{aligned} \tag{3.54}$$

Teorema 13 *Si denotamos por* $a_k = f[x_0, .., x_k]$, *entonces el polinomio de interpolación de Lagrange* $P_N(x)$ *viene dado por*

$$P_N(x) = \sum_{k=0}^{N} a_k \Pi_{i=0}^{k-1}(x - x_i), \tag{3.55}$$

donde los coeficientes $f[x_i, ..., x_k]$ *verifican*

$$f[x_i, .., x_{i+k}] = \frac{f[x_{i+1}, .., x_{i+k}] - f[x_i, .., x_{i+k-1}]}{x_{i+k} - x_i}. \tag{3.56}$$

Demostración: en primer lugar, observamos que $f[x_i, ..., x_{i+k}]$ indica, para cada $P_k(x)$, el coeficiente que acompaña a la potencia x^k en el polinomio interpolador $P_k(x)$ para los puntos $x_i, ..., x_{i+k}$. Como el polinomio interpolador es único, $f[x_i, ..., x_{i+k}]$ no depende del orden en que tomemos los puntos $x_i, ..., x_{i+k}$ y, por tanto:

$$f[x_i, ..., x_{i+k}] = f[x_{i+k}, ..., x_i]. \tag{3.57}$$

Consideremos ahora el polinomio interpolador $Q_k(x)$ que interpola en los puntos $x_{i+k}, ..., x_i$, es decir, cambiando el orden de los puntos. $Q_k(x)$ se puede escribir como

$$Q_k(x) = b_0 + b_1(x - x_{i+k}) + b_2(x - x_{k+i})(x - x_{k+i-1}) + ..., \tag{3.58}$$

donde

$$b_j = f[x_{i+k}, .., x_{i+k-j}]. \tag{3.59}$$

Por la unicidad del polinomio interpolador obtenemos que $P_k(x) = Q_k(x)$ y, por tanto

$$a_k = f[x_i, ..., x_{i+k}] = f[x_{i+k}, ..., x_i] = b_k. \tag{3.60}$$

De nuevo, por la unicidad del polinomio interpolador, los coeficientes que acompañan a la potencia x^{k-1} en ambos polinomios coinciden y, por tanto:

$$a_{k-1} - a_k \sum_{j=0}^{k-1} x_{i+j} = b_{k-1} - b_k \sum_{j=1}^{k} x_{i+j}. \tag{3.61}$$

Despejando obtenemos

$$a_k = \frac{b_{k-1} - a_{k-1}}{x_{k+i} - x_i}.$$

Finalmente obtenemos el resultado del teorema, teniendo en cuenta que

$$\begin{aligned} a_{k-1} &= f[x_i, ..., x_{i+k-1}], \\ b_{k-1} &= f[x_{i+k}...., x_{i+1}] = f[x_{i+1}...., x_{i+k}]. \end{aligned} \tag{3.62}$$

A continuación se muestra un ejemplo de cálculo del polinomio interpolador usando esta técnica tomando como puntos de interpolación $x = \{0, 2, 4, 6, 8, 10\}$ y como valores interpolados $f = \{20, 18, 16, 17, 21, 23\}$. En rojo se muestran los valores de a_k.

$$
\begin{array}{lllll}
0:20 & & & & \\
 & \frac{18-20}{2-0}=-1 & & & \\
2:18 & & \frac{-1-(-1)}{4-0}=0 & & \\
 & \frac{16-18}{4-2}=-1 & & \frac{3/8-0}{6-0}=\frac{1}{16} & \\
4:16 & & \frac{1/2-(-1)}{6-2}=\frac{3}{8} & & \frac{0-1/16}{8-0}=-\frac{1}{128} \\
 & \frac{17-16}{6-4}=\frac{1}{2} & & \frac{3/8-3/8}{8-2}=0 & \\
 & & & & \frac{-5/384-(-1/128)}{10-0}=-\frac{1}{1920} \\
6:17 & & \frac{2-1/2}{8-4}=\frac{3}{8} & & \frac{-5/48-0}{10-2}=-\frac{5}{384} \\
 & \frac{21-17}{8-6}=2 & & \frac{-1/4-3/8}{10-4}=-\frac{5}{48} & \\
8:21 & & \frac{1-2}{10-6}=-\frac{1}{4} & & \\
 & \frac{23-21}{10-8}=1 & & & \\
10:23 & & & &
\end{array}
$$

$$
P(x) = 20 - 1x + 0x(x-2) + \frac{1}{16}x(x-2)(x-4)
$$
$$
-\frac{1}{128}x(x-2)(x-4)(x-6) - \frac{1}{1920}x(x-2)(x-4)(x-6)(x-8)
$$

Descripción del algoritmo para calcular los coeficientes $a_k = f[x_0, .., x_k]$ del polinomio interpolador usando las diferencias de Newton

- Partimos de dos vectores de idéntico tamaño X_k (puntos de interpolación) y F_k (los valores interpolados).
- Creamos un vector A_k del tamaño de X_k donde almacenaremos a_k.
- Construimos un valor auxiliar B_k del tamaño de F_k y lo inicializamos a F_k. En este vector se iran actualizando los valores de las diferencias divididas.
- Hacemos $a_0 = B_0$.
- Hacemos un proceso iterativo a partir de $k = 1$. En cada iteración actualizamos todos los valores B_l usando la fórmula de las diferencias divididas
$$B_l = \frac{B_{l+1} - B_l}{X_{k+l} - X_l}$$
después de actualizar los valores B_l hacemos $A_k = B_0$.
- El criterio de parada del algoritmo se produce si en alguna iteración encontramos que X_{k+l} es igual a X_l.
- El algoritmo devuelve el vector A o un vector vacío si algo va mal.

Descripción del algoritmo para evaluar el polinomio interpolador usando las diferencias de Newton

- Partimos de dos vectores de idéntico tamaño X_k (puntos de interpolación), A_k (los coeficientes del polinomio interpolador por diferencias de Newton) y un valor x_0 donde queremos evaluar el polinomio.
- Usamos un esquema anidado similar al método de Horner descrito en el tema anterior. Según este esquema anidado la evaluación se obtiene teniendo en cuenta que si n es el grado del polinomio y definimos la secuencia :
$$b_k = A_k + b_{k+1}(x_0 - X[k]) \quad \text{con} \quad b_n = a_n,\ k = n-1, .., 0,$$
entonces la evaluación del polinomio interpolador en x_0 es b_0

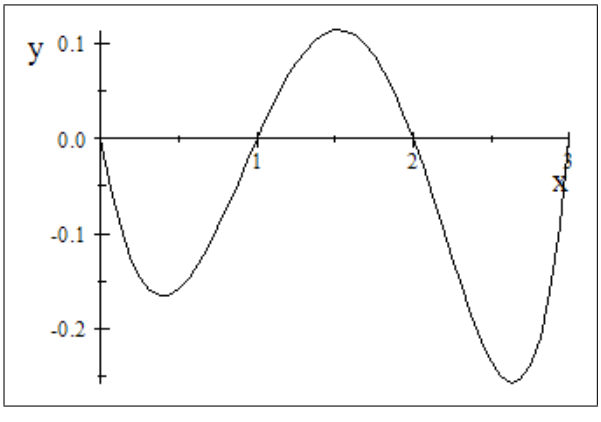
Ejemplo 7 *Sea* $f(x) = e^x$, *si interpolamos* $f(x)$ *en los puntos* $x_0 = 0$, $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, *obtenemos el polinomio interpolador de la siguiente forma:*

$$\begin{aligned} f[0,1] &= e^1 - 1, \\ f[1,2] &= e^2 - e^1, \\ f[2,3] &= e^3 - e^2, \\ f[0,1,2] &= \frac{e^2 - 2e + 1}{2}, \\ f[1,2,3] &= \frac{e^3 - 2e^2 + e^1}{2}, \\ f[0,1,2,3] &= \frac{e^3 - 3e^2 + 3e^1 - 1}{6}. \end{aligned} \tag{3.63}$$

Por tanto el polinomio interpolador de Lagrange es:

$$P_3(x) = 1 + (e-1)\,x + \frac{e^2 - 2e + 1}{2}x(x-1) + \frac{e^3 - 3e^2 + 3e^1 - 1}{6}x(x-1)(x-2). \tag{3.64}$$

En la siguiente gráfica se muestra la diferencia $e^x - P_3(x)$ en el intervalo $[0, 3]$:



Otra forma usual de aproximar funciones por polinomios son los desarrollos de Taylor. El siguiente teorema muestra el desarrollo de Taylor y su error

Teorema 14 *Sea* f *una función derivable definida en* $[a, b]$, *y* $x_0 \in [a, b]$. *Si* $x \in [a, b]$ *entonces*

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + ... \frac{f^{N)}(x_0)}{N!}(x - x_0)^N + \frac{f^{N+1)}(\xi)}{(N+1)!}(x - x_0)^{N+1},$$

donde $\xi \in [a, b]$.

Demostración: es muy similar a la demostración sobre el error del método de interpolación de Lagrange.

3.3.3. Implementación de funciones elementales.

Una vez definida una aritmética de precisión finita y las 4 operaciones básicas (suma, resta, multiplicación, división), es necesario definir, a partir de estas operaciones, las funciones elementales habituales como son: la raíz cuadrada $\sqrt{x}$, las funciones trigonométricas: $sen(x)$ $\cos(x)$ y $\tan(x)$, la función $\ln(x)$, la función e^x, la función x^y, etc. Las técnicas elementales para definir estas funciones consisten en utilizar la interpolación polinómica, los desarrollos de Taylor y los algoritmos de búsqueda de ceros (como vimos anteriormente para $\sqrt{x}$).

Aproximación de la función exponencial e^x.

Un número real x siempre se puede expresar como $x = m + x'$, donde m es un número entero y $x' \in [0, 1)$. Dado que

$$e^x = e^m e^{x'} \tag{3.65}$$

podemos descomponer el cálculo de e^x en el cálculo, por un lado, de e^m, donde al ser m un entero el cálculo es inmediato a partir de multiplicaciones sucesivas de potencias naturales de e ó e^{-1} (si $m<0$), y por otro, en el cálculo de $e^{x'}$ para $x' \in [0, 1)$. Utilizando como puntos de interpolación los asociados a los polinomios de Chebychev:

$$x_i = \frac{1}{2}\left(1 + \cos\left(\frac{2i+1}{2N+2}\pi\right)\right) \quad i = 0, ..., N, \tag{3.66}$$

obtenemos que el error relativo verifica que:

$$\frac{\mid e^{x'} - P_N(x) \mid}{e^{x'}} \leq \frac{e}{(N+1)!2^N}\left(\frac{1}{2}\right)^{N+1}. \tag{3.67}$$

Para $N = 6$, el error relativo es menor que 6.6×10^{-8} y, por tanto, del mismo orden que la unidad de redondeo u en una aritmética de 32 bits. Así, tomando un polinomio de grado $N = 6$, es decir 7 puntos de interpolación, obtenemos ya la mejor aproximación posible de e^x en el intervalo $[0, 1)$ en una aritmética de 32 bits.

Aproximación de funciones trigonométricas

Utilizaremos como modelo las funciones $f(x) = \cos(x)$ y $f(x) = sen(x)$. Puesto que estas funciones son 2π periódicas, utilizando algunas relaciones trigonométricas es suficiente definir las funciones $\cos(x)$ y $sen(x)$ en el intervalo $[0, \frac{\pi}{4}]$ y a partir de ellas definir las funciones para cualquier valor x (en radianes). Efectivamente, denotemos por $\cos_{[0,\alpha]}(x)$ y $sen_{[0,\alpha]}(x)$ a las funciones trigonométricas definidas sobre el intervalo $[0, \alpha]$. Se cumplen entonces las siguientes relaciones:

$$\cos_{[0,\frac{\pi}{2}]}(x) = \begin{cases} \cos_{[0,\frac{\pi}{4}]}(x) & si \quad x \leq \frac{\pi}{4}, \\ sen_{[0,\frac{\pi}{4}]}(\frac{\pi}{2} - x) & si \quad x > \frac{\pi}{4}. \end{cases}$$

$$sen_{[0,\frac{\pi}{2}]}(x) = \begin{cases} sen_{[0,\frac{\pi}{4}]}(x) & si \quad x \leq \frac{\pi}{4}, \\ \cos_{[0,\frac{\pi}{4}]}(\frac{\pi}{2} - x) & si \quad x > \frac{\pi}{4}. \end{cases}$$

$$\cos_{[0,\pi]}(x) = \begin{cases} \cos_{[0,\frac{\pi}{2}]}(x) & si \quad x \leq \frac{\pi}{2}, \\ -\cos_{[0,\frac{\pi}{2}]}(\pi - x) & si \quad x > \frac{\pi}{2}. \end{cases}$$

$$sen_{[0,\pi]}(x) = \begin{cases} sen_{[0,\frac{\pi}{2}]}(x) & si \quad x \leq \frac{\pi}{2}, \\ sen_{[0,\frac{\pi}{2}]}(\pi - x) & si \quad x > \frac{\pi}{2}. \end{cases}$$

$$\cos_{[0,2\pi]}(x) = \begin{cases} \cos_{[0,\pi]}(x) & si \quad x \leq \pi, \\ \cos_{[0,\pi]}(2\pi - x) & si \quad x > \pi. \end{cases}$$

$$sen_{[0,2\pi]}(x) = \begin{cases} sen_{[0,\pi]}(x) & si \quad x \leq \pi, \\ -sen_{[0,\pi]}(2\pi - x) & si \quad x > \pi. \end{cases}$$

El desarrollo en serie de Taylor centrado en 0 del $\cos(x)$ es:

$$\cos(x) \cong P_n(x) = 1 - \frac{x^2}{2} + \frac{x^4}{4!} + ... + (-1)^n \frac{x^{2n}}{(2n)!} \tag{3.68}$$

y el error máximo cometido por el desarrollo de Taylor en un punto $x \in [0, \frac{\pi}{4}]$ es

$$\mid P_n(x) - \cos(x) \mid \leq sen(x) \frac{x^{2n+1}}{(2n+1)!}. \tag{3.69}$$

La ventaja de utilizar el desarrollo de Taylor centrado en 0 es que las potencias impares de x no aparecen, lo que simplifica el cálculo numérico. El error relativo es

$$\frac{\mid P_n(x) - \cos(x) \mid}{\cos(x)} \leq \tan(x) \frac{x^{2n+1}}{(2n+1)!}. \tag{3.70}$$

Además, como $\tan(x)$ es creciente en $[0, \frac{\pi}{4}]$, el valor máximo del error se encuentra en $x = \frac{\pi}{4}$. Por ejemplo, para $n = 5$ obtenemos que el error relativo máximo cometido en $x = \frac{\pi}{4}$ es del orden de

$$\tan(\frac{\pi}{4}) \frac{\left(\frac{\pi}{4}\right)^{2 \cdot 5 + 1}}{(2 \cdot 5 + 1)!} \approx 1.8 \times 10^{-9}. \tag{3.71}$$

Por tanto, si trabajamos con una aritmética de 32 bits, cuya unidad de redondeo u es del orden de 10^{-7}, tenemos que con $n = 5$ obtenemos una aproximación del $\cos(x)$ que es la mejor posible dentro de esta aritmética y no tendría sentido aumentar el valor de n. De la misma forma es posible aproximar la función $sen(x)$.

Aproximación de la función $\ln(x)$

En primer lugar recordamos, como se vió en el tema de aritméticas de precisión finita, que los números positivos se expresan como

$$x = 2^m \left(1 + \sum_{n=1}^{t} \frac{a_n}{2^n}\right) = 2^{m+1} \left(\frac{1}{2} + \frac{1}{2} \sum_{n=1}^{t} \frac{a_n}{2^n}\right) = 2^{m+1} y.$$

donde m es un número entero e $y \in [0.5, 1)$. Aplicando las propiedades del $\ln(x)$ obtenemos que

$$\ln(x) = (m+1)\ln(2) + \ln(y). \tag{3.72}$$

Dado que el número $\ln(2)$ es una constante que supondremos calculada anteriormente ($\ln(2) \cong 0.6931471806$), podemos reducir el cálculo del $\ln(x)$ al rango de valores $0.5 \leq x < 1$.

Utilizaremos los puntos de interpolación generados por los polinomios de Chebychev, que para el intervalo $[0.5, 1]$ son:

$$x_i = \frac{1}{2} + \frac{1}{4}\left(1 + \cos\left(\frac{2i+1}{2N+2}\pi\right)\right) \quad i = 0, ..., N. \tag{3.73}$$

Dado que $\ln(1) = 0$, para minimizar el error relativo añadiremos como punto interpolante $x_{N+1} = 1$. El error de interpolación relativo entre $P_{N+1}(x)$ y $\ln(x)$ es:

$$\frac{|\ \ln(x) - P_{N+1}(x)\ |}{|\ \ln(x)\ |} = \frac{|\ (x-1)\Pi_{i=0}^{N}(x - x_i)\ |}{\xi^{N+1}(N+2)\ |\ \ln(x)\ |}, \tag{3.74}$$

donde $\xi \in [\frac{1}{2}, 1]$. Además se tiene que en el intervalo $[\frac{1}{2}, 1]$

$$\frac{|\ x-1\ |}{|\ \ln(x)\ |} \leq 1. \tag{3.75}$$

Por tanto:

$$\frac{|\ \ln(x) - P_{N+1}(x)\ |}{|\ \ln(x)\ |} \leq \frac{2^{N+1}}{N+2}\left(\frac{1}{4}\right)^{N+1} \frac{1}{2^N}. \tag{3.76}$$

Para $N = 10$ el error máximo es 3.9736×10^{-8}, que es menor que la unidad de redondeo u para una aritmética de 32 bits y por tanto tendríamos la mejor aproximación posible de la función $\ln(x)$ en dicha aritmética.

3.4. Aproximación por mínimos cuadrados (regresión lineal).

La aproximación mínimo cuadrática lineal también denominada regresión lineal en Estadística, aproxima, a través de una función, un conjunto de valores de forma global, sin exigir que la función aproximante pase exactamente por ese conjunto de puntos. Dado un conjunto de valores $\{(x_i, y_i)\}_{i=1,..,N}$, la aproximación mínimo cuadrática lineal consiste en buscar la recta $y = ax + b$, tal que la función de error cuadrático

$$E(a,b) = \sum_{i=1}^{N} (ax_i + b - y_i)^2 \tag{3.77}$$

sea mínima.

Teorema 15 *Los valores a y b que minimizan el error cuadrático anterior son*

$$\begin{aligned} a &= \frac{N\sum_{i=1}^N x_i y_i - \sum_{i=1}^N x_i \sum_{i=1}^N y_i}{N\sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i\right)^2}, \\ b &= \frac{\sum_{i=1}^N x_i^2 \sum_{i=1}^N y_i - \sum_{i=1}^N x_i y_i \sum_{i=1}^N x_i}{N\sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i\right)^2}. \end{aligned} \tag{3.78}$$

Demostración: en primer lugar, observamos que, dada la forma cuadrática que tiene el funcional, debe poseer un mínimo. Además, en un mínimo del funcional $E(a,b)$, las derivadas parciales son cero, y por tanto

$$\begin{aligned} \frac{\partial E}{\partial a}(a,b) &= 2\sum_{i=1}^{N} (ax_i + b - y_i)\, x_i = 0, \\ \frac{\partial E}{\partial b}(a,b) &= 2\sum_{i=1}^{N} (ax_i + b - y_i) = 0. \end{aligned} \tag{3.79}$$

Esto da lugar a un sistema lineal de ecuaciones cuyas incógnitas son a y b, y cuya resolución lleva al resultado establecido en el teorema.

3.5. Interpolación en 2D. Aplicación al procesado de imágenes

Una cámara digital captura la imagen en un área rectangular (el CCD) compuesta por millones de pequeños sensores cuadrados (pixels) sensibles a la luz. Podemos interpretar una imagen en blanco y negro como una función $F(x,y)$ definida sobre el area del CCD de la cual conocemos los valores en una malla de puntos $\{(x_i, y_j)\}$ (los pixels) que representan la luz captada por cada sensor del CCD. La imagen

original capturada por la cámara tiene unas dimensiones que denotaremos por $dim1 \times dim2$ pixels. Las pantallas de los dispositivos electrónicos, desde las televisiones 4K hasta los móviles, poseen resoluciones muy diferentes. Para ver la misma imagen en diferentes dispositivos electrónicos y en diferentes tamaños es necesario adaptar su resolución a la resolución de la pantalla y para ello hay que hacer lo que se denomina un zoom para modificar la resolución. Un zoom de factor z permite pasar de una imagen de dimensiones $dim1 \times dim2$ a otra de dimensiones $z \cdot dim1 \times z \cdot dim2$. Por tanto hacer un zoom es un proceso de interpolación donde tenemos que calcular los valores de la nueva imagen $\{F(x'_{i'}, y'_{j'})\}$ a partir de los valores de la imagen original $\{F(x_i, y_j)\}$.

Ilustración del procedimiento de zoom de una image. El zoom puede hacer la imagen más grande (como en este caso) o más pequeña.

A continuación vamos a estudiar como realizar un proceso de interpolación en $2D$ a partir de la interpolación en una dimensión. Concretamente utilizaremos la interpolación por el vecino más cercano y la interpolación lineal. La interpolación por el vecino más cercano es muy sencilla: dado un punto (x, y) se calcula el punto (x_i, y_i) más cercano a él y el valor interpolado es $F(x, y) \approx F(x_i, y_i)$.

En el caso de usar la interpolación lineal (que se denomina bilineal en $2D$), en primer lugar recordemos la expresión de la interpolación lineal para una variable, en el intervalo $[x_i, x_{i+1}]$, dada por

$$f(x) \approx f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i) = \left(1 - \frac{x - x_i}{x_{i+1} - x_i}\right) f(x_i) + \frac{x - x_i}{x_{i+1} - x_i} f(x_{i+1}). \tag{3.80}$$

Dada una función $F(x, y)$ de dos variables de la cual conocemos los valores en una malla de vértices $\{(x_i, y_j)\}$ y dado un punto $(x, y) \in [x_i, x_{i+1}] \times [y_j, y_{j+1}]$, si definimos

$$F_{i,j} = F(x_i, y_j), \qquad d_x^i = \frac{x - x_i}{x_{i+1} - x_i}, \qquad d_y^j = \frac{y - y_j}{y_{j+1} - y_j}, \tag{3.81}$$

aplicamos la interpolación lineal en una variable primero respecto a x y después respecto a y de la siguiente forma: primero calculamos $F(x, y_j)$ y $F(x, y_{j+1})$ interpolando respecto a x en el intervalo $[x_i, x_{i+1}]$. En segundo lugar calculamos $F(x, y)$

interpolando $F(x, y_j)$ y $F(x, y_{j+1})$ en el intervalo $[y_j, y_{j+1}]$. Lo que da lugar a las fórmulas:

$$F(x, y_j) \approx \left(1 - d_x^i\right) F_{i,j} + d_x^i F_{i+1,j}, \tag{3.82}$$

$$F(x, y_{j+1}) \approx \left(1 - d_x^i\right) F_{i,j+1} + d_x^i F_{i+1,j+1}, \tag{3.83}$$

$$F(x, y) \approx \left(1 - d_y^j\right) F(x, y_j) + d_y^j F(x, y_{j+1}), \tag{3.84}$$

combinando todo en una única fórmula obtenemos la expresión de la interpolación bilineal

$$F(x,y) \approx \left(1-d_y^j\right)\left(\left(1-d_x^i\right) F_{i,j} + d_x^i F_{i+1,j}\right) + d_y^j\left(\left(1-d_x^i\right) F_{i,j+1} + d_x^i F_{i+1,j+1}\right). \tag{3.85}$$

En el caso de las imágenes todos los puntos (x_i, y_j) están a la misma distancia entre sí y por tanto la imagen viene determinada por una matriz $F_{i,j}$ de dimensiones $dim1 \times dim2$ y los puntos tienen la forma $(x_i, y_j) = (i \cdot h, j \cdot h)$ (con $0 \leq i < dim1$ y $0 \leq j < dim2$) donde h es el tamaño del lado del pixel. Muchas veces el valor de h se desconoce y por defecto se normaliza a $h = 1$. Hacer un zoom de factor z en la imagen consiste en construir una matriz interpolada $F'_{i',j'}$ de dimensiones $(dim1 \cdot z) \times (dim2 \cdot z)$ y donde la malla de puntos tiene la forma $(x'_i, y'_j) = (i' \cdot \frac{h}{z}, j' \cdot \frac{h}{z})$ (con $0 \leq i' < dim1 \cdot z$ y $0 \leq j' < dim2 \cdot z$). Por tanto el algoritmo para hacer un zoom de factor z a una imagen usando interpolación bilineal quedaría de la siguiente forma:

Descripción algoritmo realizar un zoom de factor z a una imagen usando interpolación bilineal

- Se construye una matriz F' de dimensiones $(z \cdot dim1) \times (z \cdot dim2)$.
- Se recorren todos los puntos (i', j') de la nueva matriz.
  - Para cada (i', j') se calcula en precisión real el punto $(x, y) = (\frac{i'}{z}, \frac{j'}{z})$.
  - Se calculan los enteros $i = x$, $j = y$ (al almacenar una variable real en una entera se produce un truncamiento al eliminar los decimales).
  - Se calcula $F'_{i',j'} = F(x, y)$ usando la fórmula de interpolación bilineal. Salvo indicación de lo contrario se considera que la distancia entre los puntos de interpolación es 1. Es decir $x_{i+1} - x_i = y_{i+1} - y_i = 1$. Si al aplicar la fórmula de interpolación, $F_{i,j}, F_{i+1,j}, F_{i,j+1}$, o $F_{i+1,j+1}$ se salen de las dimensiones de la matriz F utilizamos el criterio del vecino más cercano y sustituimos dichos valores por los más cercanos dentro de la matriz.
- Se termina el proceso iterativo.

En el caso de usar la interpolación por el vecino más cercano el algoritmo es muy parecido quedando de la siguiente forma:

Descripción algoritmo realizar un zoom de factor z a una imagen usando la interpolación por el vecino más cercano

- Se construye una matriz F' de dimensiones $(z \cdot dim1) \times (z \cdot dim2)$.
- Se recorren todos los puntos (i', j') de la nueva matriz.
- Para cada (i', j') se calcula en precisión real el punto $(x, y) = (\frac{i'}{z}, \frac{j'}{z})$.
- Se calculan los enteros $i = x$, $j = y$ (al almacenar una variable real en una entera se produce un truncamiento al eliminar los decimales.)
- Se calcula que punto está más cerca de (x, y) en la cuadrícula (i, j), $(i + 1, j)$, $(i, j + 1)$ y $(i + 1, j + 1)$ y se asigna a la nueva imagen el valor de F en dicho punto.
- Si el punto más cercano cae fuera de la matriz se toma el punto dentro de la matriz más cercano.
- Se termina el proceso iterativo.

3.6. Problemas resueltos

Problema 23 *Se considera una función $f(x)$ de la cual conocemos que $f(0) = 0$, $f(1) = 3$ y $f(2) = 0$. Calcular el valor interpolado en $x = 1.2$ usando el método del vecino más próximo.*

Solución: Como el vecino más próximo es $x_i = 1$, el valor interpolado es $f(1.2) \approx 3$.

Problema 24 *Se considera una función $f(x)$ de la cual conocemos que $f(0) = 0$, $f(1) = 3$ y $f(2) = 0$. Calcular el valor interpolado en $x = 1.2$ usando el método lineal (splines de grado 1).*

Solución: Como $1.2 \in [1, 2]$ calculamos la recta que pasa por los puntos (1,1) y (2,0) obteniendo

$$f(x) = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i) = 3 + \frac{-3}{1}(x - 1),$$

por tanto en 1.2 el valor interpolado es

$$f(1.2) \approx 3 - 3(1.2 - 1) = 2.4.$$

Problema 25 *Se considera una función $f(x)$ de la cual conocemos que $f(0) = 0$, $f(1) = 3$ y $f(2) = 0$. Calcular el valor interpolado en $x = 1.2$ usando el método de splines de grado 2 y suponiendo que $f'(0) = 3$.*

Solución usando las fórmulas obtenidas para los coeficientes: De acuerdo con las fórmulas $f'(0) = b_0$ y $a_i = f(x_i)$. Por tanto, en primer lugar calculamos c_0 haciendo

$$c_0 = \frac{a_1 - a_0 - b_0(x_1 - x_0)}{(x_1 - x_0)^2} = \frac{3 - 0 - 3}{1} = 0,$$

a continuación calculamos b_1, c_1 que vienen dadas por las fórmulas

$$\begin{aligned} b_1 &= b_0 + 2c_0(x_1 - x_0) = 3 \\ c_1 &= \frac{f(x_2) - a_1 - b_1(x_2 - x_1)}{(x_2 - x_1)^2} = \frac{0 - 3 - 3}{1^2} = -6, \end{aligned}$$

de donde los polinomios interpoladores son

$$\begin{aligned} P_2^0(x) &= 3x && si \quad x \in [0, 1], \\ P_2^1(x) &= 3 + 3(x - 1) - 6(x - 1)^2 && si \quad x \in [1, 2], \end{aligned}$$

por tanto en 1.2 el valor interpolado es

$$f(1.2) \approx 3 + 3(1.2 - 1) - 6(1.2 - 1)^2 = 3.36.$$

Solución razonando a partir de las condiciones que deben cumplir los polinomios:

Los polinomios interpolantes tienen la forma

$$\begin{aligned} P_2^0(x) &= a_0 + b_0 x + c_0 x^2 && si \quad x \in [0, 1], \\ P_2^1(x) &= a_1 + b_1(x - 1) + c_1(x - 1)^2 && si \quad x \in [1, 2], \end{aligned}$$

de las condiciones que deben cumplir los polinomios vamos sacando ecuaciones de la siguiente forma:

$$\begin{aligned} P_2^0(0) = 0 \quad &\rightarrow \quad a_0 = 0, \\ \tfrac{dP_2^0}{dx}(0) = 3 \quad &\rightarrow \quad b_0 = 3, \\ P_2^0(1) = 3 \quad &\rightarrow \quad a_0 + b_0 + c_0 = 3, \\ P_2^1(1) = 3 \quad &\rightarrow \quad a_1 = 3, \\ P_2^1(2) = 0 \quad &\rightarrow \quad a_1 + b_1 + c_1 = 0, \\ \tfrac{dP_2^0}{dx}(1) = \tfrac{dP_2^1}{dx}(1) \quad &\rightarrow \quad b_0 + 2c_0 = b_1, \end{aligned}$$

y nos sale un sistema de ecuaciones que resolviendo da el mismo resultado que ya habíamos obtenido.

Problema 26 *Se considera una función $f(x)$ de la cual conocemos que $f(0) = 0$, $f(1) = 3$ y $f(2) = 0$. Calcular el valor interpolado en $x = 1.2$ usando el método de splines de grado 3 y suponiendo que $c_0 = c_2 = 0$.*

Solución: Como son solo dos intervalos vamos a hacerlo manualmente en lugar de usar las fórmulas del teorema. Como $f(0) = 0$, $f(1) = 3$ y $c_0 = 0$ los polinomios interpoladores deben ser

$$\begin{array}{ll} P_3^0(x) = b_0 x + d_0 x^3 & si \quad x \in [0,1], \\ P_3^1(x) = 3 + b_1(x-1) + c_1(x-1)^2 + d_1(x-1)^3 & si \quad x \in [1,2], \end{array}$$

y se debe cumplir:

$$\begin{array}{lll} P_3^0(1) = 3 & \rightarrow & b_0 + d_0 = 3, \\ P_3^1(2) = 0 & \rightarrow & 3 + b_1 + c_1 + d_1 = 0, \\ \frac{dP_3^0}{dx}(1) = \frac{dP_3^1}{dx}(1) & \rightarrow & b_0 + 3d_0 = b_1, \\ \frac{d^2P_3^0}{dx^2}(1) = \frac{d^2P_3^1}{dx^2}(1) & \rightarrow & 6d_0 = 2c_1, \\ c_2 = \frac{d^2P_3^1}{dx^2}(2) = 0 & \rightarrow & 2c_1 + 6d_1 = 0, \end{array}$$

nos queda un sistema de 5 ecuaciones y 5 incognitas que resolvemos y nos salen los polinomios interpoladores

$$\begin{array}{ll} P_3^0(x) = \frac{9}{2}x - \frac{3}{2}x^3 & si \quad x \in [0,1], \\ P_3^1(x) = 3 - \frac{9}{2}(x-1)^2 + \frac{3}{2}(x-1)^3 & si \quad x \in [1,2]. \end{array}$$

Si hubiesemos usado las fórmulas del teorema hubiesemos encontrado el mismo resultado. Por tanto el valor interpolado en 1.2 es

$$P_3^1(1.2) = 3 - \frac{9}{2}(1.2-1)^2 + \frac{3}{2}(1.2-1)^3 = 2.832.$$

Problema 27 *Calcular el polinomio interpolador de Lagrange $P_3(x)$ de la función $f(x) = sen(x)$ en los puntos $0, \frac{\pi}{2}, \pi$ y $\frac{3\pi}{2}$.*

Solución: Puesto que $sen(0) = sen(\pi) = 0$ sólo necesitamos los polinomios base de Lagrange centrados en $\frac{\pi}{2}$ y $\frac{3\pi}{2}$.

$$P_{\frac{\pi}{2}}(x) = \frac{x\,(x-\pi)\,\left(x-\frac{3\pi}{2}\right)}{\frac{\pi}{2}\left(\frac{\pi}{2}-\pi\right)\left(\frac{\pi}{2}-\frac{3\pi}{2}\right)},$$

$$P_{\frac{3\pi}{2}}(x) = \frac{x\,(x-\pi)\,\left(x-\frac{\pi}{2}\right)}{\frac{3\pi}{2}\left(\frac{3\pi}{2}-\pi\right)\left(\frac{3\pi}{2}-\frac{\pi}{2}\right)}.$$

Por tanto el polinomio interpolador es

$$P(x) = P_{\frac{\pi}{2}}(x) - P_{\frac{3\pi}{2}}(x).$$

Problema 28 *Se considera la función $sen(x)$ en el intervalo $[0, \frac{\pi}{8}]$ y se interpola en los puntos equidistantes $0, \frac{\pi}{32}, \frac{\pi}{16}, \frac{3\pi}{32}, \frac{\pi}{8}$. Calcular el máximo error de interpolación en dicho intervalo usando la fórmula*

$$max_{x\in[a,b]}|f(x) - P_N(x)| \leq \frac{max_{\xi\in[a,b]}|f^{N+1)}(\xi)|}{4(N+1)}h^{N+1}.$$

Solución: En este caso el número de puntos de interpolación es 5, y por tanto $N = 4$, la derivada quinta de $sen(x)$ es $\cos(x)$, cuyo valor máximo en $[0, \frac{\pi}{8}]$ es 1. Por tanto

$$max_{x\in[a,b]}|f(x) - P_N(x)| \leq \frac{1}{4\cdot 5}\left(\frac{\pi}{32}\right)^5 = 4.560048651 \times 10^{-7}.$$

Problema 29 *Se considera la función* $\cos(x)$ *en el intervalo* $[0, \frac{\pi}{8}]$ *y se interpola en los puntos equidistantes* $0, \frac{\pi}{32}, \frac{\pi}{16}, \frac{3\pi}{32}, \frac{\pi}{8}$. *Calcular el máximo error de interpolación en dicho intervalo usando la fórmula*

$$max_{x\in[a,b]}|f(x) - P_N(x)| \leq \frac{max_{\xi\in[a,b]}|f^{N+1)}(\xi)|}{4(N+1)}h^{N+1}.$$

Solución: En este caso el número de puntos de interpolación es 5, y por tanto $N = 4$, la derivada quinta de $\cos(x)$ es $sen(x)$, cuyo valor máximo en $[0, \frac{\pi}{8}]$ es $sen(\frac{\pi}{8})$. Por tanto

$$max_{x\in[a,b]}|f(x) - P_N(x)| \leq \frac{sen(\frac{\pi}{8})}{4\cdot 5}\left(\frac{\pi}{32}\right)^5 = 1.745055070 \times 10^{-7}.$$

Problema 30 *Se considera la función* $sen(x)$ *en el intervalo* $[0, \frac{\pi}{8}]$ *interpolada usando el desarrollo de Taylor centrado en* 0 *hasta el orden 4. Calcular el máximo error de interpolación en dicho intervalo usando la fórmula*

$$max_{x\in[a,b]}|f(x) - P_N(x)| \leq \frac{max_{x\in[a,b]}\left|f^{N+1)}(\xi)\right|}{(N+1)!}\left|x - x_0\right|^{N+1}.$$

Solución: En este caso $N = 4$, la derivada quinta de $sen(x)$ es $\cos(x)$, cuyo valor máximo en $[0, \frac{\pi}{8}]$ es 1. Además $x_0 = 0$ y si $x \in [0, \frac{\pi}{8}]$, $|x - x_0|$ lo más que puede valer es $\frac{\pi}{8}$. Por tanto

$$max_{x\in[a,b]}|f(x) - P_N(x)| \leq \frac{1}{5!}\left(\frac{\pi}{8}\right)^5 = 7.782483032 \times 10^{-5}.$$

Problema 31 *Interpolar la función* $f(x) = \frac{10}{x^2+1}$ *en los puntos* $x_0 = -2$, $x_1 = -1$, $x_2 = 1$, $x_3 = 2$ *utilizando las diferencias de Newton y evaluar el polinomio en* $x = 0$ *utilizando el algoritmo de Horner.*

Solución: Las diferencias de Newton se calculan de la siguiente forma:

-2 → 2			
	3		
-1 → 5		-1	
	0		0
1 → 5		-1	
	-3		
2 → 2			

Por tanto el polinomio interpolador y su evaluación en 0 se calculan de la siguiente forma:

$$P(x) = 2+3(x+2)-1(x+2)(x+1)+0(x+2)(x+1)(x-1) = (-1(x+1)+3)(x+2)+2$$
$$P(0) = (-1(0+1)+3)(0+2)+2 = 6$$

Nota: *Quitar paréntesis en P(x) y aplicar Horner sobre el polinomio resultante no es lo que pide el problema y por lo tanto está mal*

Problema 32 *Como se puede obtener la función y^x, donde x, y son números reales, utilizando las funciones e^x y $\ln(x)$.*

Solución: Se utiliza la equivalencia

$$y^x = e^{x \ln y}.$$

Problema 33 *Calcular la aproximación mínimo cuadrática lineal de la siguiente función tabulada*

x_i	y_i
0	0
1	1
2	0
3	2

Solución: Aplicando las fórmulas para calcular los coeficientes de la recta que más se aproxima a estos puntos, obtenemos:

$$a = \frac{N\sum_{i=1}^N x_i y_i - \sum_{i=1}^N x_i \sum_{i=1}^N y_i}{N\sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i\right)^2} = \frac{4(1+6)-(1+2+3)(1+2)}{4(1+2^2+3^2)-(1+2+3)^2} = \frac{1}{2},$$

$$b = \frac{\sum_{i=1}^N x_i^2 \sum_{i=1}^N y_i - \sum_{i=1}^N x_i y_i \sum_{i=1}^N x_i}{N\sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i\right)^2} = \frac{\left(1+2^2+3^2\right)(1+2)-(1+6)(1+2+3)}{4(1+2^2+3^2)-(1+2+3)^2} = 0,$$

$$P(x) = ax + b = \frac{1}{2}x.$$

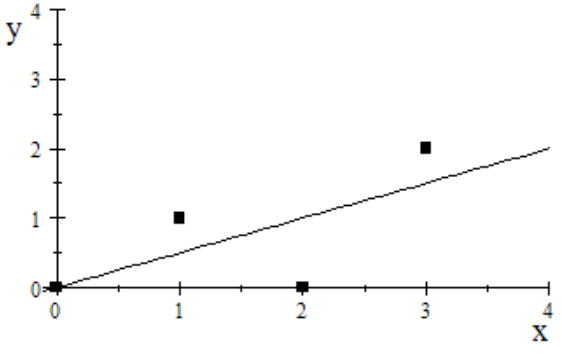


Ilustración de la aproximación mínimo cuadrática obtenida. Los cuadrados negros representan la posición de los datos utilizados.

Problema 34 *Se considera la función 2D tabulada de la siguiente forma*

$$\begin{array}{ccc} F(0,0.5)=0, & F(0,1)=1, & F(0,1.5)=2, \\ F(0.5,0.5)=3, & F(0.5,1)=4, & F(0.5,1.5)=5, \\ F(1,0.5)=6, & F(1,1)=7, & F(1,1.5)=8. \end{array}$$

Estimar $F(0.8,1.2)$ por el método del vecino más cercano y por el método de interpolación bilineal.

Solución: Como el punto de interpolación más cercano a $(0.8,1.2)$ es el punto $(1,1)$, el valor interpolado por el vecino más cercano es $F(1,1)=7$. Para la interpolación bilineal, interpolamos linealmente primero respecto a x en el intervalo $[0.5,1]$ y obtenemos

$$F(x,y) \approx \left(1-\frac{x-0.5}{0.5}\right) F(0.5,y) + \frac{x-0.5}{0.5} F(1,y).$$

A continuación interpolamos linealmente respecto a y en el intervalo $[1,1.5]$ y obtenemos

$$F(0.5,y) \approx \left(1-\frac{y-1}{0.5}\right) F(0.5,1) + \frac{y-1}{0.5} F(0.5,1.5),$$
$$F(1,y) \approx \left(1-\frac{y-1}{0.5}\right) F(1,1) + \frac{y-1}{0.5} F(1,1.5),$$

por tanto evaluando estas fórmulas para $(0.8,1.2)$ obtenemos

$$F(0.5,1.2) \approx \left(1-\frac{0.2}{0.5}\right) 4 + \frac{0.2}{0.5} 5 = 4.4,$$
$$F(1,1.2) \approx \left(1-\frac{0.2}{0.5}\right) 7 + \frac{0.21}{0.5} 8 = 7.56,$$

$$F(0.8,1.2) = \left(1-\frac{0.3}{0.5}\right) 4.4 + \frac{0.3}{0.5} 7.56 = 6.296.$$

Por tanto el valor por interpolación bilineal es 6.296.

3.7. Aplicación en Epidemiología

En este apartado, usando técnicas de interpolación, vamos a intentar responder a la siguiente pregunta : ¿Cual es la probabilidad de que una persona fallezca por la COVID-19 en función de su edad?. Para ello vamos a utilizar la información oficial comunicada por el gobierno de España hasta el día 14 de febrero de 2021 que se refleja en la siguiente tabla. Como puede observarse, la información sobre los casos se da por tramos de edad. Para pasar de la información por tramos de edad a una información más precisa donde podamos asociar a cualquier edad una aproximación de su probabilidad (o proporción) de ser hospitalizado, de que se requiera ser ingresado en UCI o de fallecer vamos a usar técnicas de interpolación.

Tabla 4. Casos de COVID-19 por nivel de gravedad notificados a la RENAVE con diagnóstico posterior al 10 de mayo de 2020. Distribución por grupo de edad

Grupo de edad (años)	Casos totales N	Hospitalizados[1] N (%)	UCI[1] N (%)	Defunciones[1] N (%)
<2	34320	854 (2,5)	30 (0,1)	16 (0,0)
2-4	55674	337 (0,6)	10 (0,0)	3 (0,0)
5-14	275969	1097 (0,4)	66 (0,0)	13 (0,0)
15-29	577280	6795 (1,2)	308 (0,1)	61 (0,0)
30-39	411607	10600 (2,6)	637 (0,2)	105 (0,0)
40-49	496923	19991 (4,0)	1704 (0,3)	374 (0,1)
50-59	434715	30159 (6,9)	3620 (0,8)	1281 (0,3)
60-69	274652	35876 (13,1)	5605 (2,0)	3452 (1,3)
70-79	175636	39550 (22,5)	4905 (2,8)	7867 (4,5)
≥80	188741	60114 (31,8)	1024 (0,5)	25587 (13,6)
Total	**2936908**	**206003 (7,0)**	**17957 (0,6)**	**38961 (1,3)**

[1] n (%) calculado sobre el total de casos en cada grupo de edad

Situación de COVID-19 en España a 24 de febrero de 2021. Equipo COVID-19. RENAVE. CNE. CNM (ISCIII)

Lo primero que vamos a hacer es fijar para cada tramo de edad una edad de referencia y supondremos que la probabilidad de esa edad de referencia coincide con la probabilidad del tramo correspondiente. Para fijar esa edad de referencia para cada tramo utilizaremos los datos que publica el Instituto Nacional de Estadística sobre la población en España que nos da, para cada edad, el número de personas, que llamaremos $N(edad)$, que hay en en España de esa edad. La edad de referencia de cada tramo la calcularemos como la media en cada tramo teniendo en cuenta $N(edad)$. Es decir, por ejemplo, para el tramo entre 5 y 14, la edad de referencia será:

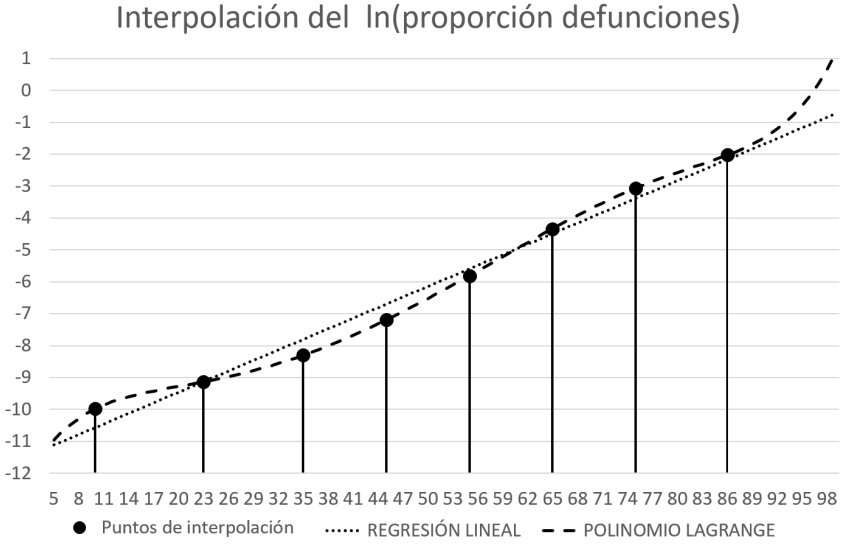
$$\text{edad referencia tramo 5-14} = \frac{\sum_{edad=5}^{edad=14} edad \cdot N(edad)}{\sum_{edad=5}^{edad=14} N(edad)}$$

en el caso del tramo de mayores de 80 años, la suma anterior termina en la última edad suministrada por el INE (100 años). Haciendo esto para cada tramo y dejando fuera los tramos de menos de 4 años, dado que el número de casos en esos tramos es poco significativo respecto a los otros tramos, obtenemos la siguiente tabla de valores donde expresamos los tramos de edad, la edad de referencia para cada tramo y las proporciones (en porcentajes) obtenidas dividiendo en cada tramo el número

de hospitalizaciones, ingresos en UCI o defunciones entre el número de casos en ese tramo.

tramo de edad	5-14	15-29	30-39	40-49	50-59	60-69	70-79	≥80
edad de refer.	10.14	22.59	35.27	45.00	54.86	64.70	74.62	86.30
% hospitalizados	0.4 %	1.2 %	2.6 %	4.0 %	6.9 %	13.1 %	22.5 %	31.8 %
% ingresos UCI	0.02 %	0.1 %	0.2 %	0.3 %	0.8 %	2.0 %	2.8 %	0.5 %
% defunciones	0.005 %	0.01 %	0.03 %	0.1 %	0.3 %	1.3 %	4.5 %	13.6 %

Para obtener una aproximación de las proporciones para cada edad se nos plantea un problema de interpolación donde los puntos de interpolación x_i son las edades de referencia y los valores de interpolación, f_i, son las proporciones. De hecho, dado que se espera una variación de tipo exponencial en las proporciones, se toman como valores de interpolación los logaritmos neperianos $ln(f_i)$. En la siguiente gráfica se muestra el resultado del proceso de interpolación en el caso del logaritmo neperiano de la proporción de defunciones usando la recta de regresión y la interpolación por el polinomio de Lagrange. Los segmentos verticales indican la posición de los puntos de interpolación y los valores interpolados.



Como puede observarse, la regresión lineal se ajusta razonablemente bien al conjunto de valores interpolados del logaritmo neperiano de la proporción de defunciones, lo que indica que efectívamente el crecimiento en la proporción de defunciones es de tipo exponencial (aunque aquí no los usaremos, hay criterios estadísticos para decidir cuando una muestra se ajusta correctamente al modelo de regresión lineal). El polinomio de Lagrange (que en este caso se calculó por el método de diferencias de Newton) pasa por todos los puntos de interpolación pero cuando nos acercamos a 100 años, $ln(f_i)$ supera el valor de cero, lo que indicaría que más del 100 % de casos fallece. Esto es, obviamente imposible y está producido por el caracter oscilante que

tienen los polinomios de grado alto, sobre todo fuera del intervalo de interpolación (en este caso el intervalo donde están los puntos de interpolación es $[10.14, 86.30]$). Por tanto concluimos que la interpolación de Lagrange no es adecuada en este caso para interpolar edades que se aproximan a 100 años y que la interpolación por regresión lineal da un resultado razonable en todas las edades. En el caso de las hospitalizaciones e ingresos en UCI podemos hacer lo mismo y podemos también probar las técnicas de interpolación por splines. En el caso particular de los ingresos en UCI, la interpolación por la regresión lineal no funciona bien porque en el último tramo de edad la proporción de ingresos en UCI baja mucho y ello produce que los datos no se aproximen bien a una recta. Nótese que estamos haciendo la interpolación sobre $ln(f_i)$, para recuperar el valor de la proporción exacta basta hacer la operación inversa. Por ejemplo, para la edad de 20 años, la regresión lineal de $ln(f_i)$ nos da el valor -9.46, por tanto la proporción de defunciones en esa edad sería $e^{-9.46} = 7.79 \times 10^{-5}$. Es decir que entre los contaminados de 20 años fallecerían 7.79 personas de cada cien mil. Esto, por supuesto es orientativo y no tiene en cuenta los factores de riesgo asociados a cada persona, como puede ser, el sexo. Por ejemplo, en el caso de la COVID-19, de cada 5 personas que fallecen, apróximadamente 3 son hombres.

Capítulo 4

ANÁLISIS NUMÉRICO MATRICIAL I

En este primer capítulo dedicado a la resolución de sistemas de ecuaciones lineales estudiaremos los métodos directos clásicos para la resolución de un sistema de ecuaciones de la forma

$$Au = b, \tag{4.1}$$

donde $A = (a_{i,j})$ es una matriz de dimensión $N \times N$, $b = (b_i)$ es un vector de tamaño N que determina el término independiente, y $u = (u_i)$ es el vector solución buscado. La condición teórica para que el sistema tenga solución única es que el determinante, $|A|$, de la matriz sea distinto de cero. En el caso en que $|A| = 0$, como muestra el teorema de Rouché–Frobenius, el sistema puede no tener ninguna solución o tener infinitas. En efecto, si $|A| = 0$ y el rango de la matriz es el mismo que el rango de la matriz ampliada (añadiendo la columna del término independiente) entonces tiene infinitas soluciones y en caso contrario ninguna. Sin embargo, como se verá más adelante, numéricamente, el valor del determinante no es un buen criterio para decidir si una matriz posee buenas cualidades a la hora de resolver numéricamente sistemas con ella.

4.1. Cálculo recursivo del determinante de una matriz

El determinante de una matriz se puede calcular como una suma de determinantes de matrices más pequeñas tal y como muestra el siguiente ejemplo:

$$|A| = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} = 1 \begin{vmatrix} 5 & 6 \\ 8 & 9 \end{vmatrix} - 2 \begin{vmatrix} 4 & 6 \\ 7 & 9 \end{vmatrix} + 3 \begin{vmatrix} 4 & 5 \\ 7 & 8 \end{vmatrix}, \tag{4.2}$$

este esquema permite una implementación recursiva de la siguiente forma:

Algoritmo recursivo para calcular el determinante de una matriz A de dimensión N:
determinante(A, N)
SI $N > 1$
construimos N matrices $A_0, ..., A_k, ...A_{N-1}$ de dimensión $N-1$. Para cada k, A_k se construye quitando la primera fila de A y su columna k-ésima.
$determinante(A, N) = \sum_{k=0}^{N-1}(-1)^k \cdot a_{0,k} \cdot determinante(A_k, N-1)$.
ELSE
determinante=$a_{0,0}$.
END

el n^o de operaciones del algoritmo recursivo determinante(A, N) es N multiplicaciones, $N-1$ sumas y N llamadas a determinantes de dimensión $N-1$. Por tanto la complejidad total de este algoritmo recursivo es de $N!$ operaciones. El factorial de un número crece tan rápido, que un algoritmo de este tipo solo es ejecutable para dimensiones de matrices pequeñas. Por ejemplo si consideramos una matriz de 100×100, que en la práctica es una matriz relativamente pequeña, calcular el determinante por este método recursivo es imposible. Efectivamente $100! \approx 10^{158}$. Uno de los super-ordenadores más rápidos que existen en la actualidad, el BlueGene/L System desarrollado por IBM, es capaz de realizar del orden de 10^{21} operaciones en como flotante por segundo. Este ordenador, para calcular un determinante de una matriz de dimensión 100 tardaría del orden de 10^{129} años. Por tanto un algoritmo de complejidad factorial con $N = 100$ no se puede ejecutar por mucho que avance la tecnología pues el número de operaciones necesarias es gigantesco e inalcanzable. Más adelante, en este capítulo veremos otras técnicas para calcular el determinante que son mucho más rápidas que este método recursivo.

4.2. Resolución de un sistema triangular de ecuaciones

Los sistemas triangulares son aquellos donde los coeficientes de la matriz son 0 de la diagonal hacia arriba (triangular inferior) o de la diagonal hacia abajo (triangular superior). Por ejemplo, una matriz triangular inferior tiene la forma :

$$\begin{pmatrix} a_{0,0} & 0 & 0 & . & 0 \\ a_{1,0} & a_{1,1} & 0 & . & 0 \\ . & . & . & . & . \\ a_{N-2,0} & . & . & a_{N-2,N-2} & 0 \\ a_{N-1,0} & a_{N-1,1} & . & . & a_{N-1,N-1} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ . \\ u_{N-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ . \\ b_{N-1} \end{pmatrix}, \quad (4.3)$$

estos sistemas se resuelven fácilmente de forma recursiva del siguiente modo:

Solución	Número de Operaciones
$u_0 = \frac{b_0}{a_{0,0}}$	1 división
$u_1 = \frac{b_1 - a_{1,0}u_0}{a_{1,1}}$	1 división + 1 suma + 1 multiplicación
$u_k = \frac{b_k - \sum_{m=0}^{k-1} a_{k,m}u_m}{a_{k,k}} \quad k=0,..,N-1$	1 división + k sumas + k multip.
Número total de operaciones	N divis.+(1+2+..+N-1) sumas y multip.

(4.4)

para valorar la complejidad computacional de los algoritmos que vamos a presentar en este capítulo utilizaremos la notación $O(a_k N^k)$ para indicar un algoritmo que requiere un número de operaciones polinomial en N y cuya potencia más alta corresponde al término $a_k N^k$. Por tanto el algoritmo anterior tiene complejidad $O(N^2)$ debido a que

$$2\,(1 + 2 + ... + N - 1) = 2\frac{N}{2}(N-1) = O(N^2), \tag{4.5}$$

Si la matriz es triangular superior se resuelve igual pero empezando de abajo hacia arriba.

4.3. Método de Gauss

Este método, aunque no es de los más rápidos, tiene la gran ventaja de que se puede aplicar a todo tipo de matrices, algo que, como veremos en el futuro, no ocurre con otros métodos más rápidos, pero que requieren, por ejemplo, que la matriz sea simétrica o definida positiva. El método de Gauss se basa en transformar el sistema $Au = b$ en un sistema equivalente $A'u = b'$ tal que la solución sea la misma y que la matriz A' sea triangular superior. Como se vio en la sección anterior, en el caso de matrices triangulares superiores, el cálculo de la solución u es inmediata siguiendo un remonte de las variables a través del siguiente esquema recursivo:

$$u_{N-1} = \frac{b'_{N-1}}{a'_{N-1,N-1}}, \tag{4.6}$$

$$u_k = \frac{b'_k - \sum_{l=k+1}^{N} a'_{k,l}u_l}{a'_{k,k}} \quad k = N-2,..,0. \tag{4.7}$$

El método de Gauss se basa en la siguiente propiedad: *cambiar el orden de las ecuaciones o sustituir una ecuación por una combinación lineal de ecuaciones no cambia la solución del sistema.*

Para obtener A' y b' se calcula, en primer lugar, el valor máximo en valor absoluto de la primera columna de A, denominado pivote. A continuación, se intercambia la primera fila de A con la fila donde se encuentra el pivote, y se hace lo mismo con el vector b, para que el sistema sea equivalente. A continuación, se multiplica la primera fila de A por el valor $\frac{-a_{k,0}}{a_{0,0}}$ y se suma a la fila $k-ésima$ de A para $k = 1, ..., N-1$. Se

hace lo mismo para el vector b, y con ello habremos obtenido un sistema equivalente tal que la primera columna es cero de la diagonal hacia abajo. Volvemos ahora a hacer lo mismo para convertir la segunda columna cero de la diagonal para abajo, y así sucesivamente hasta llegar al sistema triangular equivalente $A'u = b'$.

Ejemplo 8 *Descomposición por el método de Gauss. Se considera el sistema*

$$\begin{pmatrix} -2 & -2 & 0 \\ 6 & 18 & 12 \\ 3 & 11 & 7 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 24 \\ 8 \end{pmatrix}. \tag{4.8}$$

La descomposición del sistema se realiza en las siguientes fases:

$$\begin{pmatrix} -2 & -2 & 0 \\ 6 & 18 & 12 \\ 3 & 11 & 7 \end{pmatrix} \xrightarrow{pivoteo} \begin{pmatrix} 6 & 18 & 12 \\ -2 & -2 & 0 \\ 3 & 11 & 7 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 24 \\ 0 \\ 8 \end{pmatrix}, \tag{4.9}$$

$$\begin{pmatrix} 6 & 18 & 12 \\ -2 & -2 & 0 \\ 3 & 11 & 7 \end{pmatrix} \xrightarrow{ceros\ 1^a\ columna} \begin{pmatrix} 6 & 18 & 12 \\ 0 & 4 & 4 \\ 0 & 2 & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 24 \\ 8 \\ -4 \end{pmatrix}, \tag{4.10}$$

$$\begin{pmatrix} 6 & 18 & 12 \\ 0 & 4 & 4 \\ 0 & 2 & 1 \end{pmatrix} \xrightarrow{ceros\ 2^a\ columna} \begin{pmatrix} 6 & 18 & 12 \\ 0 & 4 & 4 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 24 \\ 8 \\ -8 \end{pmatrix}, \tag{4.11}$$

y el remonte da como solución $u_2 = 8$, $u_1 = \frac{8-4u_2}{4} = -6$, $u_0 = \frac{24-18u_1-12u_2}{6} = 6$.

Recuento de Operaciones del método de Gauss

Convertir en 0 el elemento $a_{1,0}$	1 división + N multip. y sumas
Convertir en cero $a_{1,0}, a_{2,0}, ...a_{N-1,0}$	N-1 divis. + N(N-1) multip.y sumas
Convertir en cero $a_{2,1}, a_{3,1}, ...a_{N-1,1}$	N-2 divis. + (N-1)(N-2) multip. y sumas
Convertir en cero $a_{N-1,N-2}$	1 divis. + 2 multip. y sumas
Total Operaciones :	(1+2+..+N-1) divisiones
	+ (2+6+...+N(N-1)) multip. y sumas

La complejidad del método de Gauss es $O(\frac{2}{3}N^3)$ debido a que la parte significativa del cómputo de operaciones viene dada por

$$2\,(2 + 6 + ... + N(N-1)) = 2\frac{N^3 - N}{3} = O(\frac{2}{3}N^3). \tag{4.12}$$

El algoritmo del método de Gauss se puede implementar con pivotación física donde cada vez que se pivotan filas se intercambian todos los elementos, o con pivotación virtual donde solo se guarda un registro de los cambios de filas que se van haciendo. A continuación se muestra el algoritmo con pivotación virtual:

Descripción algoritmo método de Gauss con pivotación física

- Se inicia un proceso iterativo donde en cada iteración se hace cero la matriz de la diagonal hacia abajo. En cada iteración, k, se hace lo siguiente
  - Se calcula donde se encuentra el máximo en valor absoluto de la diagonal hacia abajo de la matriz y se intercambia esa fila con la fila de la diagonal (lo mismo con el término independiente)
  - Para cada fila i de la diagonal hacia abajo se calcula

    $$m = \frac{a_{i,k}}{a_{k,k}}$$

    y la fila i se actualiza haciendo

    $$a_{i,j} = a_{i,j} - m \cdot a_{k,j}$$

    y se hace lo mismo para el término independiente
  - El criterio de parada es que el máximo de la diagonal hacia abajo sea cero.
- Una vez convertido el sistema en triangular se llama a la función remonte para calcular la solución.

Descripción algoritmo método de Gauss con pivotación virtual

Se define e inicializa a la identidad el vector piv para gestionar la pivotación.

El algoritmo funciona igual que el de pivotación física con solo tres cambios:

- Cuando se calcula el máximo de la diagonal hacia abajo, en lugar de hacer una pivotación física, simplemente se intercambian los valores de piv en las posiciones de las filas correspondientes
- Cuando se accede a una posición de la matriz, siempre se usa $A[piv[i]][j]$ en lugar de $A[i][j]$ (lo mismo para el vector independiente)
- El algoritmo de remonte para resolver el sistema tiene que tener en cuenta la pivotación virtual

En el caso de la pivotación física no existe el vector piv[.] y cada vez que es necesario pivotar filas se intercambian los valores de las filas y los correspondientes del vector b. Si durante el proceso de triangularización de A, en algún momento $a'_{i,i}$ es cero, el algoritmo, en principio, no puede continuar. Sin embargo, si el último elemento de la matriz $a'_{N-1,N-1} = 0$ y $b'_{N-1} = 0$ entonces el sistema tiene infinitas

soluciones y podemos fijar una solución haciendo, por ejemplo, $u_{N-1} = 1$ y calculando por un remonte el resto de incógnitas.

Uso del método de Gauss para calcular la inversa de una matriz

Una variante del método de Gauss se puede usar para calcular la inversa A^{-1} de una matriz. Para ello tenemos en cuenta la igualdad

$$\underbrace{\begin{pmatrix} a_{0,0} & . & a_{0,N-1} \\ . & . & . \\ a_{N-1,0} & . & a_{N-1,N-1} \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u_{0,0} & . & u_{0,N-1} \\ . & . & . \\ u_{N-1,0} & . & u_{N-1,N-1} \end{pmatrix}}_{A^{-1}} = \underbrace{\begin{pmatrix} 1 & . & 0 \\ . & . & . \\ 0 & . & 1 \end{pmatrix}}_{B}. \quad (4.13)$$

Usando el método de Gauss haciendo las mismas operaciones en A que en la matriz B llegamos a

$$\underbrace{\begin{pmatrix} a_{0,0} & . & a_{0,N-1} \\ 0 & . & . \\ 0 & 0 & a_{N-1,N-1} \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u_{0,0} & . & u_{0,N-1} \\ . & . & . \\ u_{N-1,0} & . & u_{N-1,N-1} \end{pmatrix}}_{A^{-1}} = \underbrace{\begin{pmatrix} b_{0,0} & . & b_{0,N-1} \\ . & . & . \\ b_{N-1,0} & . & b_{N-1,N-1} \end{pmatrix}}_{B}. \quad (4.14)$$

Por tanto, podemos calcular A^{-1} por columnas, teniendo en cuenta que cada columna k se puede calcular resolviendo el sistema triangular

$$\underbrace{\begin{pmatrix} a_{0,0} & . & a_{0,N-1} \\ 0 & . & . \\ 0 & 0 & a_{N-1,N-1} \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u_{0,k} \\ . \\ u_{N-1,k} \end{pmatrix}}_{u_k} = \underbrace{\begin{pmatrix} b_{0,k} \\ . \\ b_{N-1,k} \end{pmatrix}}_{b_k}. \quad (4.15)$$

Respecto a la complejidad computacional, para convertir la matriz A en triangular son necesarias $O(\frac{2}{3}N^3)$ operaciones. Para realizar las operaciones necesarias en la matriz B al mismo tiempo que la triangulación de A hay que tener en cuenta que cada vez que hacemos 0 un elemento no diagonal de A tenemos que combinar dos filas de la matriz B multiplicando una fila por un número y sumando el resultado a otra fila, lo cual nos da $2N$ operaciones. Como el número de elementos por debajo de la diagonal de A es $\frac{N^2-N}{2}$ ellos nos da un total de $N^3 - N^2$ operaciones para modificar B. A continuación hay que resolver N sistemas triangulares cada uno de los cuales requiere $O(N^2)$ operaciones, por tanto el cómputo total de operaciones para calcular A^{-1} por este método es

$$O(\frac{2}{3}N^3) + O(N^3) + O(N^3) = O(\frac{8}{3}N^3).$$

Descripción algoritmo método de Gauss para calcular la inversa con pivotación física

Se define e inicializa a la identidad una matriz B del tamaño de A.

El algoritmo funciona igual que el de pivotación física para resolver sistemas teniendo en cuenta lo siguiente:

- Todas las operaciones que se hacían para cada elemento del vector b ahora se hacen para las filas de la matriz B.
- El algoritmo de remonte para resolver el sistema tiene que tener en cuenta que utiliza, en lugar del vector b, la matriz B y que devuelve la matriz solución. En nuestro caso la matriz inversa.

4.4. Método de Cholesky

Este método sólo se puede aplicar a matrices simétricas y definidas positivas. El siguiente teorema da 3 posibles definiciones equivalentes de una matriz definida positiva.

Teorema 16 *Sea A una matriz simétrica, las 3 siguientes afirmaciones son definiciones equivalentes de que la matriz sea definida positiva*

(*i*) $\forall \bar{x} \neq 0 \;\; {}^t\bar{x} A \bar{x} > 0$.

(*ii*) *Todos los autovalores λ de A son positivos.*

(*iii*) *Los determinantes de todos los menores principales de A son positivos.*

El método de Cholesky se basa en descomponer la matriz A en la forma:

$$A = B \cdot {}^tB, \tag{4.16}$$

donde B es una matriz triangular inferior.

$$B = \begin{pmatrix} b_{0,0} & 0 & 0 & . & 0 \\ b_{1,0} & b_{1,1} & 0 & . & . \\ b_{2,0} & b_{2,1} & b_{2,2} & . & . \\ . & . & . & . & 0 \\ b_{N-1,1} & b_{N-1,2} & b_{N-1,3} & . & b_{N-1,N-1} \end{pmatrix}, \tag{4.17}$$

realizando el producto $B \cdot {}^tB$ e igualándolo a A se puede deducir el siguiente algoritmo para calcular B:

Algoritmo factorización Cholesky

Para $i = 0, ..., N-1$

$b_{i,i} = \sqrt{a_{i,i} - \sum_{k=0}^{i-1} b_{i,k}^2}$

Para $j = i+1, ..., N-1$

$b_{j,i} = \frac{1}{b_{i,i}} \left(a_{j,i} - \sum_{k=0}^{i-1} b_{j,k} b_{i,k} \right)$

Fin Para j

Fin Para i

El interés de descomponer una matriz A por el método de Cholesky es que, a continuación, es muy sencillo resolver el sistema de ecuaciones $Au = b$. Efectivamente, basta descomponer el sistema de la siguiente forma:

$$\begin{aligned} Bz &= b, \\ {}^tBu &= z. \end{aligned} \tag{4.18}$$

Ambos sistemas se resuelvan rápidamente haciendo un remonte y un descenso.

Nota: Numéricamente resulta conveniente almacenar B y ${}^tB$ en una única matriz simétrica, escribiendo en la parte triangular superior de B la parte correspondiente a ${}^tB$.

Respecto a la complejidad computacional, se puede comprobar que la parte significativa corresponde a la descomposición de Cholesky de la matriz B que tiene un total de $O(\frac{1}{3}N^3)$ operaciones que es más rápido que el método de Gauss que tiene una complejidad de $O(\frac{2}{3}N^3)$.

4.5. Factorización LU de una matriz

La factorización LU es un método parecido al de Cholesky que sirve para matrices no-simétricas. Busca una descomposición de la forma :

$$A = \begin{pmatrix} 1 & 0 & . & 0 \\ l_{1,0} & 1 & . & 0 \\ . & . & . & . \\ l_{N-1,0} & l_{N-1,1} & . & 1 \end{pmatrix} \begin{pmatrix} u_{0,0} & u_{0,1} & . & u_{0,N-1} \\ 0 & u_{1,1} & . & u_{1,N-1} \\ . & . & . & . \\ 0 & 0 & . & u_{N-1,N-1} \end{pmatrix}. \tag{4.19}$$

Por ejemplo :

$$\begin{pmatrix} 3 & 2 & 1 \\ -3 & 0 & -2 \\ 6 & 2 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -1 & 1 \end{pmatrix} \begin{pmatrix} 3 & 2 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.20}$$

Multiplicando las matrices L y U e igualando a la matriz A se obtiene el siguiente algoritmo para obtener $A = L \cdot U$:

Algoritmo factorización *LU*
Para $i = 0, ..., N-1$
  $l_{i,i} = 1$
  $u_{i,i} = a_{i,i} - \sum_{k=0}^{i-1} l_{i,k} u_{k,i}$
  Para $j = i+1, ..., N-1$
    $u_{i,j} = a_{i,j} - \sum_{k=0}^{i-1} l_{i,k} u_{k,j}$
    $l_{j,i} = \frac{1}{u_{i,i}} \left(a_{j,i} - \sum_{k=0}^{i-1} l_{j,k} u_{k,i} \right)$
  Fin Para j
Fin Para i

una vez calculadas L y U, podemos utilizarlas para resolver sistemas de ecuaciones. Si tenemos el sistema $Au = L \cdot U \cdot u = b$ podemos resolverlo descomponiéndolo en los siguientes sistemas triangulares :

$$\begin{aligned} Lz &= b, \\ Uu &= z. \end{aligned} \tag{4.21}$$

Ambos sistemas se resuelven rápidamente haciendo un remonte y un descenso. Se puede comprobar que la descomposición LU tiene una complejidad igual que la del método de Gauss, es decir $O(\frac{2}{3}N^3)$.

La factorización LU no funciona para cualquier tipo de matrices. Por ejemplo, si intentamos aplicar la descomposición LU a la matriz

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{4.22}$$

que tiene $|A| = -1 \neq 0$, se obtiene que debe cumplirse la igualdad

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l_{1,0} & 1 \end{pmatrix} \begin{pmatrix} u_{0,0} & u_{0,1} \\ 0 & u_{1,1} \end{pmatrix} = \begin{pmatrix} u_{0,0} & u_{0,1} \\ l_{1,0}u_{0,0} & l_{1,0}u_{0,1} + u_{1,1} \end{pmatrix}, \tag{4.23}$$

lo cual es imposible porque $u_{0,0} = 0$ y por tanto $l_{1,0}u_{0,0} = 0 \neq 1$. Para que la factorización LU funcione para cualquier tipo de matrices con $|A| \neq 0$ habría que perfeccionarlo añadiendo una matriz de permutaciones (véase por ejemplo [In-Re]).

4.6. Método de Crout para matrices tridiagonales

El caso de sistemas de ecuaciones con matrices A tridiagonales posee una forma especialmente simple de factorización. Vamos a descomponer A en el producto de dos matrices triangulares de la forma siguiente:

$$\begin{pmatrix} a_0 & b_0 & . & 0 \\ c_0 & a_1 & . & 0 \\ 0 & . & . & b_{N-2} \\ 0 & . & c_{N-2} & a_{N-1} \end{pmatrix} = \begin{pmatrix} l_0 & 0 & . & 0 \\ m_0 & l_1 & . & 0 \\ 0 & . & . & 0 \\ 0 & . & m_{N-2} & l_{N-1} \end{pmatrix} \begin{pmatrix} 1 & u_0 & . & 0 \\ 0 & 1 & . & 0 \\ 0 & . & . & u_{N-2} \\ 0 & . & 0 & 1 \end{pmatrix}, \tag{4.24}$$

multiplicando las matrices triangulares e igualando a la matriz original se deduce fácilmente el siguiente algoritmo para calcular los coeficientes m_i, l_i, y u_i.

Algoritmo factorización de Crout

$l_0 = a_0$

$u_0 = \frac{b_0}{l_0}$

Para $i = 1, .., N-2$

$m_{i-1} = c_{i-1}$

$l_i = a_i - m_{i-1}u_{i-1}$

$u_i = \frac{b_i}{l_i}$

Fin Para

$m_{N-2} = c_{N-2}$

$l_{N-1} = a_{N-1} - m_{N-2}u_{N-2}$

Finalmente, el sistema de ecuaciones se resuelve mediante un descenso y un remonte tal y como se hace en las otras factorizaciones. En este caso es más sencillo dada la estructura que tiene la descomposición. Se puede comprobar fácilmente (ver la lista de problemas) que la complejidad computacional del método de Crout es $O(8N)$.

En la siguiente tabla se muestra un resumen de los diferentes métodos que se han visto en este tema ordenados desde el más rápido hacia el más lento.

método	tipo matrices	complejidad
Crout	tridiagonales con $\|A\| \neq 0$	$O(8N)$
Triangulares	triangulares con $\|A\| \neq 0$	$O(N^2)$
Cholesky	simétricas y definidas positivas	$O(\frac{1}{3}N^3)$
Factorización LU	matrices que sean factorizables	$O(\frac{2}{3}N^3)$
Gauss	cualquier matriz con $\|A\| \neq 0$	$O(\frac{2}{3}N^3)$
Inversa con Gauss	cualquier matriz con $\|A\| \neq 0$	$O(\frac{8}{3}N^3)$

4.7. Estimación del error de un método para resolver sistemas.

Para estimar la fiabilidad de la solución numérica de un sistema de ecuaciones, haremos lo siguiente: dada una matriz A, un vector de términos independientes b y un vector solución u, calculado utilizando alguna técnica numérica, si la solución es perfecta entonces $Au - b = 0$. Ahora bien, esto puede no suceder porque los errores de redondeo y de cálculo producen que esta estimación no sea exacta. Para estimar el error cometido al resolver el sistema utilizaremos la expresión siguiente, donde e es el vector $e = Au - b$:

$$ErrorSistema = \frac{1}{N}\sum \frac{|e_i|}{|b_i| + \epsilon}, \tag{4.25}$$

donde N es la dimensión del sistema, ϵ es un número que se pone para evitar posibles divisiones por cero. *ErrorSistema* representa el error relativo medio al resolver el sistema. Cuanto más pequeño sea *ErrorSistema*, mejor aproximada estará la solución del sistema.

4.8. Cálculo del determinante mediante factorización o triangularización de matrices

Dadas dos matrices A, B de dimensión N, sus determinantes verifican

$$|A \cdot B| = |A| \cdot |B|,$$

además si A es triangular entonces

$$|A| = \prod_{k=0}^{N-1} a_{k,k}.$$

Aplicando esto a las factorizaciones de Cholesky, LU y Crout obtenemos

- Cholesky : $A = B \cdot {}^tB \ \rightarrow \ |A| = \left(\prod_{k=0}^{N-1} b_{k,k}\right)^2$.
- Factorización LU : $A = L \cdot U \ \rightarrow \ |A| = \prod_{k=0}^{N-1} U_{k,k}$.
- Factorización Crout : $A = L \cdot U \ \rightarrow \ |A| = \prod_{k=0}^{N-1} l_k$.

También es posible calcular el determinante de una matriz usando el procedimiento del método de Gauss para convertir una matriz cualquiera en una matriz triangular superior. Para ello se usan las siguientes dos propiedades :

1. Al pivotar dos filas de una matriz el determinante de la matriz cambia de signo.
2. La operación del procedimiento del método de Gauss que consiste en sustituir las filas por debajo de la diagonal por la suma de la propia fila y la multiplicación de la fila diagonal por un número no cambia el valor del determinante de la matriz.

Teniendo en cuenta estas dos propiedades, si A' es la matriz triangular obtenida por el método de Gauss a partir de una matriz A, entonces

$$|A| = (-1)^{N_{piv}} \prod_{k=0}^{N-1} a'_{k,k} \tag{4.26}$$

done N_{piv} es el número de pivotaciones realizadas en el procedimiento del método de Gauss y $a'_{k,k}$ son los elementos diagonales de la matriz triangular A'.

Calcular el determinante usando estos métodos es mucho más rápido que usar el algoritmo recursivo explicado al principio del tema que tiene una complejidad factorial.

4.9. Problemas resueltos

Problema 35 *Resolver por el método de Gauss el sistema*

$$\begin{pmatrix} -1 & 2 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}. \tag{4.27}$$

Solución: Hacemos una pivotación, convertimos en 0 el elemento por debajo de la diagonal y resolvemos el sistema triangular:

$$\begin{pmatrix} -1 & 2 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \end{pmatrix} \rightarrow \tag{4.28}$$

$$\begin{pmatrix} 2 & -1 \\ 0 & \frac{3}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \end{pmatrix} \rightarrow \frac{3}{2}y = 3 \rightarrow y = 2 \rightarrow x = \frac{2}{2} = 1. \tag{4.29}$$

Problema 36 *Resolver por el método de Gauss el siguiente sistema de ecuaciones*

$$\begin{pmatrix} -1 & 2 & 2 \\ 2 & 4 & 4 \\ 0 & 2 & 2 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 18 \\ 6 \end{pmatrix}. \tag{4.30}$$

Solución: Pasos en la descomposición por Gauss:

1. Usamos la matriz ampliada y pivotamos la primera y segunda fila:

$$\begin{pmatrix} -1 & 2 & 2 & 3 \\ 2 & 4 & 4 & 18 \\ 0 & 2 & 2 & 6 \end{pmatrix} \xrightarrow{pivoteo} \begin{pmatrix} 2 & 4 & 4 & 18 \\ -1 & 2 & 2 & 3 \\ 0 & 2 & 2 & 6 \end{pmatrix}. \tag{4.31}$$

2. Hacemos ceros en la primera columna

$$\begin{pmatrix} 2 & 4 & 4 & 18 \\ -1 & 2 & 2 & 3 \\ 0 & 2 & 2 & 6 \end{pmatrix} \xrightarrow{ceros} \begin{pmatrix} 2 & 4 & 4 & 18 \\ 0 & 4 & 4 & 12 \\ 0 & 2 & 2 & 6 \end{pmatrix}. \tag{4.32}$$

3. Hacemos ceros en la segunda columna

$$\begin{pmatrix} 2 & 4 & 4 & 18 \\ 0 & 4 & 4 & 12 \\ 0 & 2 & 2 & 6 \end{pmatrix} \xrightarrow{ceros} \begin{pmatrix} 2 & 4 & 4 & 18 \\ 0 & 4 & 4 & 12 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{4.33}$$

4. Como en el sistema final $a_{2,2} = b_2 = 0$ el sistema tiene infinitas soluciones. Para elegir una fijamos el valor de $u_2 = 1$ y realizamos un remonte para calcular las otras incógnitas:

$$u_2 = 1, \qquad u_1 = \frac{12 - 4u_2}{4} = 2, \qquad u_0 = \frac{18 - 4u_2 - 4u_1}{2} = 3. \tag{4.34}$$

Problema 37 *Demostrar que si $A = B \cdot {}^tB$ (B triangular inferior) y $|B| \neq 0$, entonces A es simétrica y definida positiva.*

Solución: tenemos que demostrar, por una parte, que ${}^tA = A$ (A simétrica) y, por otra, que $\forall \bar{x} \neq 0 \ {}^t\bar{x}A\bar{x} > 0$ (A definida positiva)

1. Simétrica: ${}^tA = {}^t\big(B \cdot {}^tB\big) = B \cdot {}^tB = A$.

2. Definida positiva ($\forall \bar{x} \neq 0, {}^t\bar{x}A\bar{x} > 0$) :

$$ {}^t\bar{x}A\bar{x} = {}^t\bar{x}B \cdot {}^tB\bar{x} = {}^t\big({}^tB\bar{x}\big) \cdot {}^tB\bar{x}. $$

Por tanto llamando $\bar{y} = {}^tB\bar{x}$ se obtiene que ${}^t\bar{x}A\bar{x} = {}^t\bar{y}\bar{y} = \sum y_i^2$.

Por último, si $|B| \neq 0$ y $\bar{x} \neq 0$ entonces $\bar{y} = {}^tB\bar{x} \neq 0$ y por tanto

$$ {}^t\bar{x}A\bar{x} = {}^t\bar{y}\bar{y} = \sum y_i^2 > 0 $$

.

Problema 38 *Descomponer la siguiente matriz A por el método de Cholesky*

$$ A = \begin{pmatrix} 1 & 1 & 4 \\ 1 & 5 & 6 \\ 4 & 6 & 26 \end{pmatrix}. \tag{4.35} $$

Solución: La descomposición por el método de Cholesky tiene la forma siguiente:

$$ A = B \cdot {}^tB, \tag{4.36} $$

donde la matriz B es triangular inferior. Por tanto haciendo este producto obtenemos

$$ \begin{pmatrix} b_{00} & 0 & 0 \\ b_{10} & b_{11} & 0 \\ b_{20} & b_{21} & b_{22} \end{pmatrix} \begin{pmatrix} b_{00} & b_{10} & b_{20} \\ 0 & b_{11} & b_{21} \\ 0 & 0 & b_{22} \end{pmatrix} = \begin{pmatrix} b_{00}^2 & b_{00}b_{10} & b_{00}b_{20} \\ b_{00}b_{10} & b_{10}^2 + b_{11}^2 & b_{10}b_{20} + b_{11}b_{21} \\ b_{00}b_{20} & b_{10}b_{20} + b_{11}b_{21} & b_{20}^2 + b_{21}^2 + b_{22}^2 \end{pmatrix}. \tag{4.37} $$

Igualamos los elementos de la matriz anterior con los elementos de la matriz A y se obtienen los siguientes resultados:

$$ \begin{array}{ccc} b_{00}^2 = 1 & \rightarrow & b_{00} = 1, \\ b_{00}b_{10} = 1 & \rightarrow & b_{10} = \frac{1}{b_{00}} = 1, \\ b_{00}b_{20} = 4 & \rightarrow & b_{20} = \frac{4}{b_{00}} = 4, \\ b_{10}^2 + b_{11}^2 = 5 & \rightarrow & b_{11} = \sqrt{5 - b_{10}^2} = \sqrt{4} = 2, \\ b_{10}b_{20} + b_{11}b_{21} = 6 & \rightarrow & b_{21} = \frac{6 - b_{10}b_{20}}{b_{11}} = \frac{6-4}{2} = 1, \\ b_{20}^2 + b_{21}^2 + b_{22}^2 = 26 & \rightarrow & b_{22} = \sqrt{26 - b_{20}^2 - b_{21}^2} = \sqrt{26 - 16 - 1^2} = 3. \end{array} \tag{4.38} $$

La descomposición queda de la siguiente manera:

$$A = B \cdot {}^tB = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 4 & 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 4 \\ 0 & 2 & 1 \\ 0 & 0 & 3 \end{pmatrix}. \tag{4.39}$$

Problema 39 *Calcular el determinante de la siguiente matriz usando la descomposición LU*

$$A = \begin{pmatrix} 2 & -1 & -2 \\ 2 & 2 & 3 \\ -8 & 7 & 17 \end{pmatrix}. \tag{4.40}$$

Solución: La descomposición LU consiste en expresar A como

$$A = \begin{pmatrix} 1 & 0 & 0 \\ l_{1,0} & 1 & 0 \\ l_{2,0} & l_{2,1} & 1 \end{pmatrix} \begin{pmatrix} u_{0,0} & u_{0,1} & u_{0,2} \\ 0 & u_{1,1} & u_{1,2} \\ 0 & 0 & u_{2,2} \end{pmatrix}, \tag{4.41}$$

multiplicando sale la condición

$$\begin{pmatrix} 2 & -1 & -2 \\ 2 & 2 & 3 \\ -8 & 7 & 17 \end{pmatrix} = \begin{pmatrix} u_{0,0} & u_{0,1} & u_{0,2} \\ l_{1,0}u_{0,0} & l_{1,0}u_{0,1} + u_{1,1} & l_{1,0}u_{0,2} + u_{1,2} \\ l_{2,0}u_{0,0} & l_{2,0}u_{0,1} + l_{2,1}u_{1,1} & l_{2,0}u_{0,2} + l_{2,1}u_{1,2} + u_{2,2} \end{pmatrix}, \tag{4.42}$$

si vamos despejando progresivamente van saliendo todos los coeficientes de L y U obteniendo

$$\begin{pmatrix} 2 & -1 & -2 \\ 2 & 2 & 3 \\ -8 & 7 & 17 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -4 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 & -2 \\ 0 & 3 & 5 \\ 0 & 0 & 4 \end{pmatrix}, \tag{4.43}$$

finalmente, como el determinante del producto de matrices es el producto de los determinantes y el determinante de una matriz triangular es el producto de los elementos diagonales obtenemos

$$\left| \begin{pmatrix} 2 & -1 & -2 \\ 2 & 2 & 3 \\ -8 & 7 & 17 \end{pmatrix} \right| = (1 \cdot 1 \cdot 1)(2 \cdot 3 \cdot 4) = 24. \tag{4.44}$$

Problema 40 *Deducir el algoritmo de Crout para factorizar matrices tridiagonales.*

Solución: Consideremos la matriz tridiagonal siguiente:

$$A = \begin{pmatrix} a_0 & b_0 & 0 & \cdots & 0 \\ c_0 & a_1 & b_1 & \cdots & 0 \\ 0 & c_1 & a_2 & b_2 & 0 \\ \vdots & \vdots & \vdots & \ddots & b_{N-2} \\ 0 & 0 & 0 & c_{N-2} & a_{N-1} \end{pmatrix}. \tag{4.45}$$

La descomposición por el método de Crout genera dos matrices de la forma:

$$A = \begin{pmatrix} l_0 & 0 & . & 0 \\ m_0 & l_1 & . & 0 \\ 0 & . & . & 0 \\ 0 & . & m_{N-2} & l_{N-1} \end{pmatrix} \begin{pmatrix} 1 & u_0 & . & 0 \\ 0 & 1 & . & 0 \\ 0 & . & . & u_{N-2} \\ 0 & . & 0 & 1 \end{pmatrix} = \tag{4.46}$$

$$= \begin{pmatrix} l_0 & l_0u_0 & 0 & . & 0 \\ m_0 & m_0u_0 + l_1 & l_1u_1 & . & 0 \\ 0 & . & . & . & l_{N-2}u_{N-2} \\ 0 & 0 & . & m_{N-2} & m_{N-2}u_{N-2} + l_{N-1} \end{pmatrix}. \tag{4.47}$$

Igualando ambas matrices y despejando los elementos l_i, u_i y m_i obtenemos

$$\begin{array}{l} l_0 = a_0 \\ u_0 = \frac{b_0}{l_0} \\ \text{Para i} = 1, \ldots, N-2 \\ \qquad m_{i-1} = c_{i-1} \\ \qquad l_i = a_i - m_{i-1}u_{i-1} \\ \qquad u_i = \frac{b_i}{l_i} \\ \text{Fin Para} \\ m_{N-2} = c_{N-2} \\ l_{N-1} = a_{N-1} - m_{N-2}u_{N-2} \end{array} \tag{4.48}$$

Problema 41 *Resolver utilizando el método de Crout el siguiente sistema de ecuaciones*

$$\begin{pmatrix} 2 & 4 & 0 \\ -1 & 0 & 4 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 6 \\ 3 \\ -1 \end{pmatrix}. \tag{4.49}$$

Solución: Para calcular la factorización de Crout escribimos:

$$\begin{pmatrix} 2 & 4 & 0 \\ -1 & 0 & 4 \\ 0 & -1 & 0 \end{pmatrix} = \begin{pmatrix} l_0 & 0 & 0. \\ m_0 & l_1 & 0 \\ 0 & m_1 & l_2 \end{pmatrix} \begin{pmatrix} 1 & u_0 & 0 \\ 0 & 1 & u_1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} l_0 & l_0u_0 & 0 \\ m_0 & l_1 + m_0u_0 & l_1u_1 \\ 0 & m_1 & l_2 + m_1u_1 \end{pmatrix}$$

de donde despejando obtenemos

$l_0 = 2$	$u_0 = \frac{4}{2} = 2$	
$m_0 = -1$	$l_1 = 0 - 2(-1) = 2$	$u_1 = \frac{4}{2} = 2$
$m_1 = -1$	$l_2 = 0 - 2(-1) = 2$	

Sustituyendo estos valores en las matrices de Crout, la descomposición queda:

$$A = L \cdot U = \begin{pmatrix} 2 & 0 & 0 \\ -1 & 2 & 0 \\ 0 & -1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}, \tag{4.50}$$

Para resolver el sistema, se utiliza un descenso y un remonte haciendo

$$\begin{aligned} Ly &= b, \\ Ux &= y. \end{aligned} \tag{4.51}$$

Calculamos el valor de y a partir del sistema anterior:

$$\begin{pmatrix} 2 & 0 & 0 \\ -1 & 2 & 0 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 3 \\ -1 \end{pmatrix}, \tag{4.52}$$

aplicando un algoritmo de descenso,

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \frac{6}{2} \\ \frac{3+y_1}{2} \\ \frac{-1+y_2}{2} \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix} \tag{4.53}$$

Calculamos el vector x por remonte haciendo $Ux = y$

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix}, \tag{4.54}$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 - 2x_2 \\ 3 - 2x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{4.55}$$

Problema 42 *Calcular el número de operaciones necesarias para resolver un sistema tridiagonal por el método de Crout.*

Solución: De acuerdo con el algoritmo de factorización de Crout las operaciones que se hacen dentro del bucle principal del algoritmo son :

$$\begin{aligned} m_{i-1} &= c_{i-1} \\ l_i &= a_i - m_{i-1}u_{i-1} \\ u_i &= \frac{b_i}{l_i} \end{aligned} \tag{4.56}$$

es decir una resta, una multiplicación y una división, por tanto el orden de operaciones del bucle es $3N$. A continuación se resuelve el sistema haciendo primero un descenso para resolver $Lz = b$ donde la principal operación en el bucle principal es

$$z_i = \frac{b_i - m_{i-1}z_{i-1}}{l_i}$$

que de nuevo da un orden de complejidad de $3N$. Finalmente se realiza un remonte para resolver $Ux = z$ donde la principal operación en el bucle principal es

$$x_i = z_i - u_{i+1}x_{i+1}$$

que da un orden de complejidad de $2N$. Por tanto, la complejidad del algoritmo de Crout para resolver sistemas es $O(8N)$.

4.10. Aplicación en Epidemiología

En este apartado vamos a hacer una simulación de como se va propagando el COVID-19 entre los diferentes grupos de edad partiendo de un único paciente cero que va contaminando a otra personas. Llamaremos generación 0 a la formada por el mencionado paciente cero, llamaremos generación 1 a las personas que contamina este paciente 0, llamaremos generación 2 a los que contaminan los de la generación 1, y así sucesivamente, supóndremos que estamos al principio de la epidemía donde no hay implantadas medidas de distanciamiento social y supóndremos que la tasa de reproducción es $R = 3$ es decir que cada persona contaminada contamina (en media) a 3 personas. Para estudiar como se va propagando a través de los grupos de edad vamos a utilizar la llamada matriz de contactos sociales que llamaremos A, que es una aproximación de cuantos contactos diarios se producen entre los diferentes grupos de edad (organizados en 16 grupos de 5 años). Utilizaremos la siguiente matriz de contactos para España publicada en [Prem]:

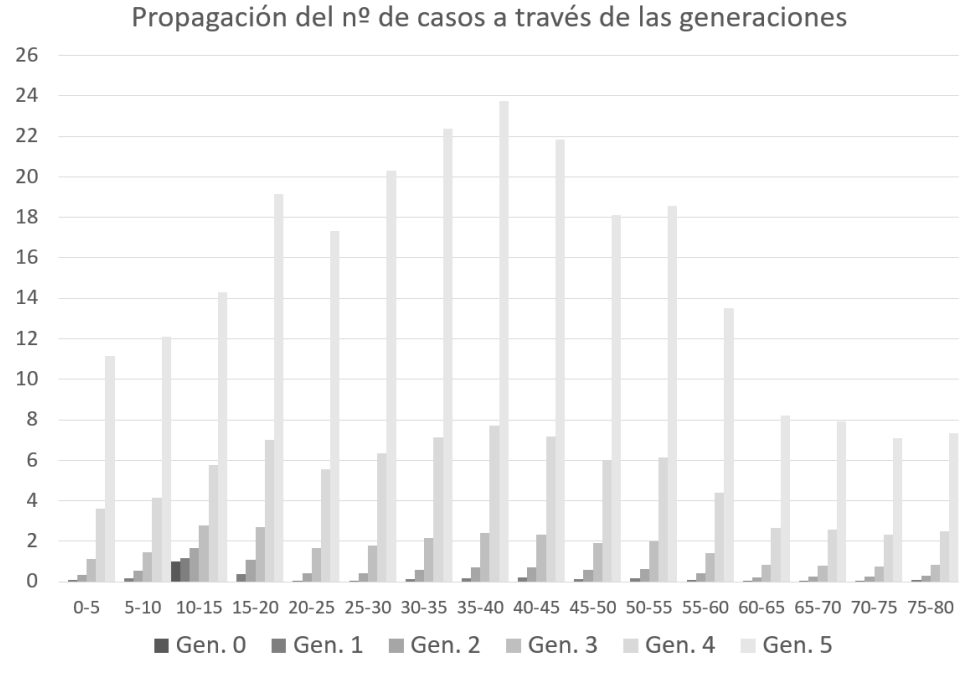
AÑOS	0-5	5-10	10-15	15-20	20-25	25-30	30-35	35-40	40-45	45-50	50-55	55-60	60-65	65-70	70-75	75-80	SUMA
0-5	3.978	0.955	0.331	0.208	0.230	0.502	0.992	1.058	0.658	0.312	0.276	0.218	0.155	0.139	0.079	0.055	10.15
5-10	0.908	4.807	0.805	0.208	0.141	0.341	0.692	0.911	0.952	0.372	0.206	0.162	0.139	0.115	0.052	0.054	10.86
10-15	0.184	1.352	5.651	0.577	0.218	0.217	0.375	0.662	0.960	0.512	0.290	0.126	0.074	0.084	0.066	0.068	11.42
15-20	0.101	0.241	1.741	6.013	1.117	0.606	0.545	0.687	0.865	0.917	0.513	0.164	0.055	0.048	0.028	0.028	13.67
20-25	0.133	0.156	0.203	1.663	3.280	1.625	1.216	1.132	0.919	1.119	0.812	0.372	0.081	0.047	0.051	0.053	12.86
25-30	0.358	0.241	0.145	0.721	1.921	3.587	2.016	1.598	1.312	1.080	1.081	0.586	0.182	0.078	0.036	0.038	14.98
30-35	0.735	0.919	0.577	0.455	1.070	1.966	3.629	2.281	1.645	1.213	0.908	0.610	0.317	0.155	0.075	0.082	16.64
35-40	0.878	1.157	0.811	0.656	0.772	1.539	2.136	3.704	2.349	1.414	0.945	0.456	0.319	0.245	0.130	0.067	17.58
40-45	0.421	0.841	0.933	1.000	0.895	1.300	1.798	2.083	3.099	1.674	1.106	0.343	0.191	0.160	0.109	0.068	16.02
45-50	0.417	0.529	0.590	1.410	0.883	1.022	1.347	1.534	1.585	2.194	1.100	0.415	0.149	0.109	0.095	0.118	13.50
50-55	0.226	0.573	0.821	1.142	0.994	1.287	1.255	1.250	1.592	1.706	1.903	0.709	0.240	0.143	0.096	0.122	14.06
55-60	0.432	0.569	0.491	0.651	0.602	1.069	1.180	0.908	0.964	0.807	0.961	1.346	0.438	0.240	0.111	0.112	10.88
60-65	0.349	0.298	0.194	0.311	0.296	0.511	0.779	0.770	0.524	0.403	0.383	0.579	1.080	0.424	0.233	0.122	7.26
65-70	0.238	0.345	0.267	0.158	0.232	0.394	0.766	0.744	0.634	0.371	0.388	0.512	0.551	1.135	0.293	0.170	7.20
70-75	0.101	0.276	0.304	0.285	0.150	0.287	0.369	0.596	0.679	0.488	0.385	0.321	0.603	0.632	0.924	0.326	6.73
75-80	0.233	0.316	0.465	0.379	0.164	0.207	0.413	0.489	0.563	0.642	0.628	0.390	0.283	0.410	0.354	0.619	6.55
SUMA	9.69	13.58	14.33	15.84	12.96	16.46	19.51	20.41	19.30	15.23	11.89	7.31	4.86	4.16	2.73	2.10	

La suma por columnas de la matriz representa el total de contactos que genera un grupo de edad. Por ejemplo el tramo de edad entre 35-40 años es el grupo que genera más contactos (un total de 20.41). Esto se debe a que este grupo tiene contactos familiares (con sus hijos y con sus padres), tiene contactos laborales y tiene contactos sociales (reuniones amigos, etc..). Según nos hacemos mayores, cada vez tenemos menos contactos y el grupo de 75-80 años tiene muchos menos contactos (un total de 2.10). Esta matriz de contactos es válida cuando no hay medidas de distanciamiento social y las personas se relacionan libremente. Vamos a denotar por u^n al vector de tamaño 16 que representa el número de contaminados en la generación n distribuidos por grupos de edad. Para calcular u^{n+1} a partir de u^n supóndremos que u^{n+1} es proporcional al vector $c^n = Au^n$, es decir, supóndremos que los contagios que genera cada grupo de edad es propocional a los contactos que genera dicho grupo. Ahora bien, como hemos supuesto que en cada generación se multiplica por $R = 3$ el número

de contaminados, tenemos que escalar c^n para que se cumpla esto, es decir que la suma de contagios en u^{n+1} sea R veces la suma de contagios en u^n, de donde nos sale la fórmula:

$$u^{n+1} = R\frac{\sum_{k=1}^{16} u_k^n}{\sum_{k=1}^{16} c_k^n} c^n, \tag{4.57}$$

en la gráfica siguiente se muestra una simulación de las 5 primeras generaciones obtenidas partiendo de un paciente 0 en la franja de 10-15 años. Se observa que con una tasa de reproducción de $R = 3$ en la quinta generación, el número de contaminados es $243 = 3^5$, encontrándose el mayor número de contaminados en la franja entre 35-40 años.



En este contexto, se plantean de forma natural 2 cuestiones:

1. ¿ Con el avance de las generaciones, la distribución de casos entre franjas de edad depende de la franja de edad del paciente cero ?

2. ¿A partir de los datos de una generación, es posible volver hacia atrás y encontrar la franja de edad del paciente cero?

la respuesta a la primera pregunta es que la distribución por franjas de edad va convergiendo hacia una única distribución independientemente de la franja de edad del paciente cero. De hecho, converge hacia el autovector máximo de la matriz A. Esto se verá cuando se estudie en el tema 6 los autovalores y autovectores de matrices. Para responder a la segunda pregunta basta tener en cuenta que podemos invertir

la ecuación (4.57) para pasar de u^{n+1} a u^n a través de la fórmula

$$u^n = \frac{1}{R}\frac{\sum_{k=1}^{16} u_k^{n+1}}{\sum_{k=1}^{16} z_k^n} z^n, \tag{4.58}$$

donde z es la solución del sistema $Az = u^{n+1}$. Por tanto, teóricamente, es posible ir hacia atrás; pero en la práctica, lo que sucede numéricamente es que si ya han avanzado muchas generaciones, los datos de contaminados están muy mezclados por franjas de edad y la propagación de errores en la resolución del sistema para pasar de una generación a la anterior produce que al final no lleguemos correctamente a identificar la franja de edad del paciente cero. Esto dependerá del número de generaciones que se haya avanzado y de la precisión de la aritmética.

Capítulo 5

DIFERENCIACIÓN E INTEGRACIÓN NUMÉRICA

Una fórmula de diferenciación numérica es un procedimiento que permite aproximar una derivada de una función $f(x)$ en un punto x_i utilizando el valor de $f(x)$ en otros puntos vecinos a x_i. Por otro lado, una fórmula de integración numérica es un procedimiento que permite aproximar el valor de la integral en un intervalo $[a, b]$ a partir de la evaluación de $f(x)$ en algunos puntos incluidos en el intervalo $[a, b]$. A pesar de que desde un punto de vista teórico la derivación e integración son temas muy relacionados, numéricamente son problemas completamente distintos. La derivación es algo local, es decir interviene el punto y vecinos cercanos y la integración es global pues interviene una región entera (por ejemplo un intervalo en el caso de dimensión 1).

5.1. Diferenciación Numérica

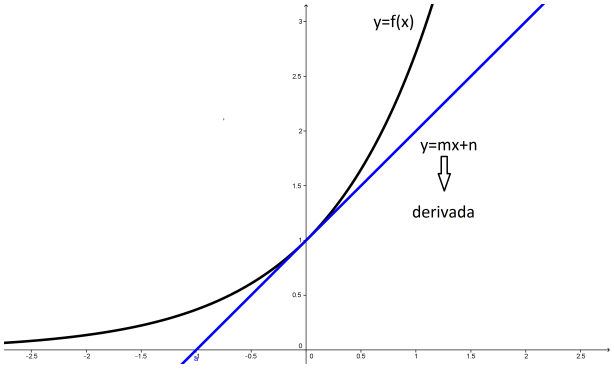


Ilustración de la recta tangente a una función en $x = 0$

Geométricamente la derivada de una función $f(x)$ en un punto x es la pendiente de la recta tangente en dicho punto. Una derivada positiva indica que la función tiende a crecer en dicho punto y negativa indica que tiende a decrecer.

5.1.1. Aproximación de la derivada a través de un límite

La definición formal de derivada viene dada por el siguiente límite

$$f'(x) = \underset{h\to 0}{Lim}\frac{f(x+h)-f(x)}{h}, \tag{5.1}$$

por tanto, esta fórmula nos da una primera forma de aproximar la derivada pues si h es un valor pequeño, entonces podemos aproximar la derivada haciendo

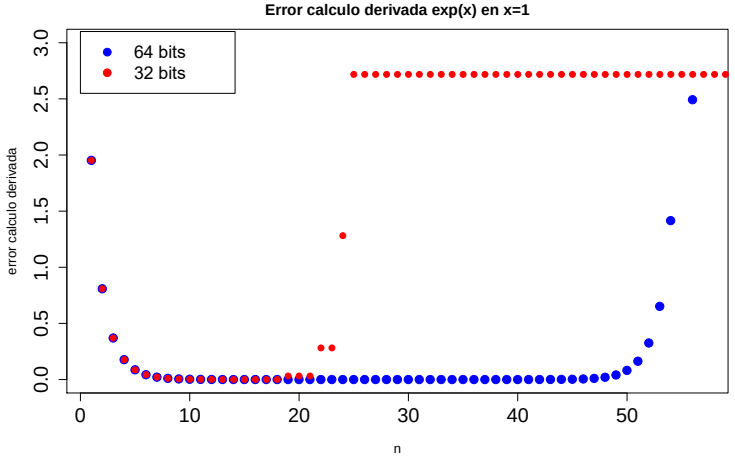
$$f'(x) \approx \frac{f(x+h)-f(x)}{h}. \tag{5.2}$$

Cuando trabajamos con una aritmética de precisión finita hay que tener en cuenta que el valor h que tiene sentido tomar no es cualquiera, el valor de h está condicionado por la calidad de la aritmética (32 bits, 64 bits, etc..) y por la magnitud del número x. En el siguiente ejemplo se ilustra este comportamiento para la derivada de $f(x) = e^x$ en $x = 1$. Aplicando la fórmula anterior obtenemos:

$$f'(1) = e \approx \frac{e^{1+h}-e}{h}, \tag{5.3}$$

en la gráfica y tablas que siguen se ilustra el valor del error en la estimación tomando $h = \frac{1}{2^n}$.

$$E(n) = Error\left(\frac{1}{2^n}\right) = \left|\frac{e^{1+\frac{1}{2^n}}-e}{\frac{1}{2^n}} - e\right|. \tag{5.4}$$



Valores del error $E(n)$ dado por (5.4) en una aritmética de 32 bits.

n	$E(n)$	n	$E(n)$	n	$E(n)$	n	$E(n)$	n	$E(n)$	n	$E(n)$
1	2	2	$8{\cdot}10^{-1}$	4	$2{\cdot}10^{-1}$	6	$2{\cdot}10^{-1}$	8	$1{\cdot}10^{-2}$	10	$3{\cdot}10^{-3}$
12	$1{\cdot}10^{-3}$	14	$5{\cdot}10^{-4}$	16	$5{\cdot}10^{-4}$	18	$5{\cdot}10^{-4}$	20	$3{\cdot}10^{-2}$	22	$3{\cdot}10^{-1}$
24	1										

Valores del error $Error(n)$ dado por (5.4) en una aritmética de 64 bits.

n	$E(n)$	n	$E(n)$	n	$E(n)$	n	$E(n)$	n	$E(n)$	n	$E(n)$
1	2	4	$2{\cdot}10^{-1}$	8	$1{\cdot}10^{-2}$	12	$7{\cdot}10^{-4}$	16	$4{\cdot}10^{-5}$	20	$3{\cdot}10^{-6}$
24	$2{\cdot}10^{-7}$	28	$2{\cdot}10^{-8}$	32	$4{\cdot}10^{-7}$	36	$5{\cdot}10^{-6}$	40	$1{\cdot}10^{-4}$	44	$8{\cdot}10^{-4}$
48	$1{\cdot}10^{-2}$	52	$4{\cdot}10^{-2}$								

Como puede observarse en la gráfica para la aritmética de 32 bits (que tiene una precisión para la mantisa de $t = 23$ bits), cuando n es pequeño, o cuando n se acerca a $n = 23$ que es la precisión máxima de la mantisa se produce un fuerte error en la estimación. En el caso de n pequeño el error se produce porque el $h = \frac{1}{2^n}$ asociado es muy grande y cuando h se aproxima a $h = \frac{1}{2^{23}}$ el error se produce porque se está desbordando la capacidad de la aritmética de precisión finita. En el caso de la aritmética de 64 bits ($t = 52$) se puede tomar un h mucho más pequeño sin que se desborde la aritmética y además se puede conseguir un menor error en la estimación de la derivada. Además, tanto para 32 como para 64 bits se obtiene que el error menor se obtiene alrededor de la zona central del rango de valores de $n \in [1, t]$. Si $u = \frac{1}{2^t}$ es la unidad de redondeo, el valor de h asociado a dicho valor central es $\sqrt{u}$. Una elección habitual de h, teniendo en cuenta además la magnitud del número x donde se quiere calcular la derivada es

$$h \approx (|x| + \epsilon)\sqrt{u}, \tag{5.5}$$

donde $\epsilon \geq 0$ es un número pequeño que se puede poner cuando $|x| \approx 0$. En el caso en que sea conocido el intervalo $[a, b]$ en que se mueve el valor de x, por ejemplo si x mide ángulos sabemos que se mueve en $[0, 2\pi]$, entonces podemos tomar un único h para todo el intervalo dado por

$$h = max\{|a|, |b|\}\sqrt{u}. \tag{5.6}$$

5.1.2. Aproximación de la derivada a través de los desarrollos de Taylor

La manera habitual de aproximar las derivadas de una función $f(x)$ en un punto x_i consiste en utilizar el desarrollo de Taylor centrado en x_i:

$$f(x) = f(x_i) + \frac{f'(x_i)}{1!}(x - x_i) + \frac{f''(x_i)}{2!}(x - x_i)^2 ... + \frac{f^{N)}(x_i)}{N!}(x - x_i)^N + ... \tag{5.7}$$

Si tomamos un punto $x = x_j \neq x_i$, y despejamos, obtenemos la siguiente expresión:

$$f'(x_i) = \frac{f(x_j) - f(x_i)}{x_j - x_i} - \frac{f''(x_i)}{2!}(x_j - x_i) ... - \frac{f^{N)}(x_i)}{N!}(x_j - x_i)^{N-1} - ... = \tag{5.8}$$

$$= \frac{f(x_j) - f(x_i)}{x_j - x_i} + O\left(|x_j - x_i|\right), \tag{5.9}$$

en general, $O\left(|x_j - x_i|^k\right)$ indica que el error cometido es una suma de potencias de $|x_j - x_i|$ en la que la potencia más pequeña es k. k determina el orden de aproximación. Cuanto mayor sea k mayor precisión tendrá la fórmula de aproximación de la derivada. Si $x_j > x_i$, entonces la derivada se calcula hacia adelante, mientras que si $x_j < x_i$, la derivada se calcula hacia atrás. Nótese que haciendo $x_j = x_i + h$ obtenemos la fórmula obtenida en el apartado anterior.

Para obtener una fórmula con mayor precisión vamos a combinar la fórmula anterior en dos puntos x_l y x_r. Es decir

$$f'(x_i) = \frac{f(x_l) - f(x_i)}{x_l - x_i} - \frac{f''(x_i)}{2!}(x_l - x_i)... - \frac{f^{N)}(x_i)}{N!}(x_l - x_i)^{N-1} - ... \tag{5.10}$$
$$f'(x_i) = \frac{f(x_r) - f(x_i)}{x_r - x_i} - \frac{f''(x_i)}{2!}(x_r - x_i)... - \frac{f^{N)}(x_i)}{N!}(x_r - x_i)^{N-1} - ...,$$

multiplicando la primera fórmula por $(x_r - x_i)$, la segunda por $(x_l - x_i)$ y restándolas se obtiene que se elimina el término que contiene $f''(x_i)$, por tanto despejando $f'(x_i)$ se obtiene la fórmula:

$$f'(x_i) = \frac{(x_i - x_l)\frac{f(x_r)-f(x_i)}{x_r-x_i} + (x_r - x_i)\frac{f(x_i)-f(x_l)}{x_i-x_l}}{x_r - x_l} + O(h^2), \tag{5.11}$$

donde $h = max\{|\ x_r - x_i\ |, |\ x_l - x_i\ |\}$. Nótese que, si $x_r = x_i + h$, y $x_l = x_i - h$, entonces la fórmula anterior se simplifica de la siguiente forma:

$$f'(x_i) = \frac{f(x_i + h) - f(x_i - h)}{2h}, \tag{5.12}$$

que es una conocida fórmula de diferencias centradas.

En la siguiente tabla se muestra una comparación del resultado del cálculo de la derivada de la función $f(x) = x^3 + x$ en $x = 1$ usando las fórmula de error $O(h)$ y la de error $O(h^2)$ (el resultado exacto es $f'(1) = 4$).

	Fórmula $f'(x) = \frac{f(x+h)-f(x)}{h} + O(h)$	*Fórmula* $f'(x) = \frac{f(x+h)-f(x-h)}{2h} + O(h^2)$
$h = 1$	$f'(1) \approx \frac{f(2)-f(1)}{1} = 8$	$f'(1) \approx \frac{f(2)-f(0)}{2} = 5$
$h = 0.1$	$f'(1) \approx \frac{f(1.1)-f(1)}{0.1} = 4.31$	$f'(1) \approx \frac{f(1.1)-f(0.9)}{0.2} = 4.01$
$h = 0.01$	$f'(1) \approx \frac{f(1.01)-f(1)}{0.01} = 4.03$	$f'(1) \approx \frac{f(1.01)-f(0.99)}{0.02} = 4.0001$

Como puede apreciarse, cuando se divide por 10 la distancia h, el error se divide apróximadamente por 10 en el caso de la fórmula de error $O(h)$ y por 10^2 en el caso de la fórmula de error $O(h^2)$.

5.1.3. Fórmulas para calcular la derivada segunda

Partimos de nuevo de los desarrollos en serie de Taylor

$$f(x_r) = f(x_i) + \frac{f'(x_i)}{1!}(x_r - x_i) + \frac{f''(x_i)}{2!}(x_r - x_i)^2 + \frac{f'''(x_i)}{3!}(x_r - x_i)^3 + ... \quad (5.13)$$

$$f(x_l) = f(x_i) + \frac{f'(x_i)}{1!}(x_l - x_i) + \frac{f''(x_i)}{2!}(x_l - x_i)^2 + \frac{f'''(x_i)}{3!}(x_l - x_i)^3 + ..., \quad (5.14)$$

ahora hay que combinar los dos desarrollos para eliminar el término en $f'(x_i)$. Para ello multiplicamos la primera ecuación por $(x_l - x_i)$, la segunda por $(x_r - x_i)$, las restamos y despejamos $f''(x_i)$ obteniendo:

$$f''(x_i) = 2\frac{\frac{f(x_r)-f(x_i)}{x_r-x_i} - \frac{f(x_i)-f(x_l)}{x_i-x_l}}{x_r - x_l} + O(h). \quad (5.15)$$

En el caso de que los puntos sean equidistantes, es decir $x_r = x_i + h$ y $x_l = x_i - h$, la fórmula para calcular la segunda derivada se simplifica y además se cancelan los términos $f'''(x_i)$ del desarrollo obteniendo

$$f''(x_i) = \frac{f(x_i + h) + f(x_i - h) - 2f(x_i)}{h^2} + O(h^2). \quad (5.16)$$

5.1.4. Diferenciación numérica en 2D. Aplicación al procesado de imágenes

Estudiaremos, en este apartado, la aproximación de las derivadas de una función de varias variables. Para simplificar la exposición, supondremos que la dimensión es 2. Una primera aproximación de las derivadas parciales se obtiene utilizando las fórmulas obtenidas en dimensión 1 en cada variable dejando la otra variable constante. Por ejemplo, si usamos la fórmula para calcular la derivada de orden $O(h^2)$, las derivadas parciales de una función $F(x, y)$ se pueden aproximar como

$$\frac{\partial F(x,y)}{\partial x} = \frac{F(x + h, y) - F(x - h, y)}{2h} + O(h^2), \quad (5.17)$$

$$\frac{\partial F(x,y)}{\partial y} = \frac{F(x, y + h) - F(x, y - h)}{2h} + O(h^2). \quad (5.18)$$

Para obtener fórmulas más precisas de las derivadas de una función $F(x, y)$, se utilizan los desarrollos de Taylor siguientes en 2 variables. Utilizaremos la siguiente nomenclatura: $F_x = \frac{\partial F(x,y)}{\partial x}$, $F_y = \frac{\partial F(x,y)}{\partial y}$, $F_{xx} = \frac{\partial^2 F(x,y)}{\partial x^2}$, $F_{xy} = \frac{\partial^2 F(x,y)}{\partial x \partial y}$, $F_{yy} = \frac{\partial^2 F(x,y)}{\partial y^2}$.

1. $F(x + h, y) = F + hF_x + \frac{h^2}{2}F_{xx} + O(h^3)$,

2. $F(x-h,y) = F - hF_x + \frac{h^2}{2}F_{xx} + O(h^3)$,
3. $F(x,y+l) = F + lF_y + \frac{l^2}{2}F_{yy} + O(l^3)$,
4. $F(x,y-l) = F - lF_y + \frac{l^2}{2}F_{yy} + O(l^3)$,
5. $F(x+h,y+l) = F + hF_x + lF_y + \frac{1}{2}(h^2F_{xx} + 2hlF_{xy} + l^2F_{yy}) + O(\left(h^2+l^2\right)^{\frac{3}{2}})$,
6. $F(x-h,y-l) = F - hF_x - lF_y + \frac{1}{2}(h^2F_{xx} + 2hlF_{xy} + l^2F_{yy}) + O(\left(h^2+l^2\right)^{\frac{3}{2}})$,
7. $F(x+h,y-l) = F + hF_x - lF_y + \frac{1}{2}(h^2F_{xx} - 2hlF_{xy} + l^2F_{yy}) + O(\left(h^2+l^2\right)^{\frac{3}{2}})$,
8. $F(x-h,y+l) = F - hF_x + lF_y + \frac{1}{2}(h^2F_{xx} - 2hlF_{xy} + l^2F_{yy}) + O(\left(h^2+l^2\right)^{\frac{3}{2}})$.

Prestaremos particular atención a dos operadores diferenciales que se utilizan con frecuencia en la práctica: El gradiente $\nabla F(x,y) = (F_x(x,y), F_y(x,y))$, que es el vector de derivadas parciales, y el Laplaciano $\Delta F(x,y) = F_{xx}(x,y) + F_{yy}(x,y)$.

Supondremos que la función $F(x,y)$ está discretizada y utilizaremos la notación $F_{i,j} \cong F(hi,lj)$. Para expresar de forma compacta el cálculo de las diferentes derivadas en 2 variables se usan las denominadas máscaras de convolución 3×3. Una máscara de convolución viene determinada por una matriz 3×3 de coeficientes

$$M = \begin{array}{|c|c|c|}\hline m_{0,0} & m_{0,1} & m_{0,2} \\ \hline m_{1,0} & m_{1,1} & m_{1,2} \\ \hline m_{2,0} & m_{2,1} & m_{2,2} \\ \hline \end{array}, \tag{5.19}$$

la aplicación de una máscara M a una función discretizada $F_{i,j}$ se denomina $M * F$ y es a su vez una función discretizada que se calcula como

$$M * F_{i,j} = \sum_{k=-1}^{1}\sum_{l=-1}^{1} m_{k+1,l+1}F_{i+k,j+l}. \tag{5.20}$$

Aplicar la máscara M en un punto (i,j) de una función tabulada $F_{i,j}$ se puede interpretar como poner la matriz M encima de $F_{i,j}$ centrándola en la posición (i,j) e ir multiplicando y sumando los coeficientes de la máscara por los valores de la función que quedan debajo de la máscara. Por ejemplo, aplicar la máscara

$$M = \begin{array}{|c|c|c|}\hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \end{array}, \tag{5.21}$$

corresponde a hacer una media de un punto y sus 8 valores vecinos. Dados a, b números reales, al aplicar la máscara

$$M_x = \begin{array}{|c|c|c|}\hline -b & 0 & b \\ \hline -a & 0 & a \\ \hline -b & 0 & b \\ \hline \end{array}, \tag{5.22}$$

se obtiene

$$M_x * F_{i,j} = a(F_{i+1,j} - F_{i-1,j}) + b(F_{i+1,j+1} - F_{i-1,j+1} + F_{i+1,j-1} - F_{i-1,j-1}), \quad (5.23)$$

utilizando el desarrollo de Taylor y despejando se obtiene

$$F_x(hi, lj) = \frac{M_x * F_{i,j}}{(2a + 4b)h} + O((h^2 + l^2)^{\frac{1}{2}}), \quad (5.24)$$

para calcular los parámetros a, b se impone en primer lugar que

$$(2a + 4b)h = 1, \quad (5.25)$$

para que al aplicar la máscara M_x obtengamos directamente la derivada en la dirección horizontal. Para completar el cálculo de a y b hay diferentes estrategias. Aquí vamos a usar la de imponer que a, b sean inversamente proporcionales a la distancia del punto donde se aplica al punto central, (la distancia del punto central a su vecino de la derecha es h y la distancia del punto central a su vecino diagonal es $\sqrt{h^2 + l^2}$). En nuestro caso esa condición nos lleva a

$$\frac{a}{b} = \frac{\sqrt{h^2 + l^2}}{h}, \quad (5.26)$$

uniendo las dos condiciones y resolviendo el sistema asociado obtenemos:

$$a = \frac{\sqrt{h^2 + l^2}}{h\left(4h + 2\sqrt{h^2 + l^2}\right)}, \qquad b = \frac{1}{4h + 2\sqrt{h^2 + l^2}}, \quad (5.27)$$

haciendo lo mismo para el cálculo de la derivada vertical obtenemos

$$M_y = \begin{array}{|c|c|c|} \hline -d & -c & -d \\ \hline 0 & 0 & 0 \\ \hline d & c & d \\ \hline \end{array}, \quad (5.28)$$

de donde se deduce usando los desarrollos de Taylor:

$$F_y(hi, lj) = \frac{M_y * F_{i,j}}{(2c + 4d)l} + O((h^2 + l^2)^{\frac{1}{2}}), \quad (5.29)$$

c y d se calculan usando la misma estrategia que a, b obteniendo:

$$c = \frac{\sqrt{h^2 + l^2}}{l\left(4l + 2\sqrt{h^2 + l^2}\right)}, \qquad d = \frac{1}{4l + 2\sqrt{h^2 + l^2}}. \quad (5.30)$$

Para implementar la máscara del laplaciano M_Δ también hay diferentes estrategias. Aquí vamos a usar una máscara sencilla que viene dada por

$$M_\Delta = \begin{array}{|c|c|c|} \hline 0 & \frac{1}{l^2} & 0 \\ \hline \frac{1}{h^2} & -\frac{2}{l^2} - \frac{2}{h^2} & \frac{1}{h^2} \\ \hline 0 & \frac{1}{l^2} & 0 \\ \hline \end{array}, \quad (5.31)$$

se puede comprobar, usando los desarrollos de Taylor, que

$$\Delta F(hi, lj) = M_\Delta * F_{ij} + O(h^2 + l^2). \tag{5.32}$$

Como puede observarse, todas las derivadas se calculan por aplicación de diferentes máscaras 3×3. En el caso de que $h = l$ (es decir la misma distancia entre los puntos en vertical y horizontal). Las máscaras anteriores se simplifican y quedan

$$M_x = \frac{1}{4h} \begin{array}{|c|c|c|} \hline -(2-\sqrt{2}) & 0 & (2-\sqrt{2}) \\ \hline -2(\sqrt{2}-1) & 0 & 2(\sqrt{2}-1) \\ \hline -(2-\sqrt{2}) & 0 & (2-\sqrt{2}) \\ \hline \end{array}, \tag{5.33}$$

$$M_y = \frac{1}{4h} \begin{array}{|c|c|c|} \hline (2-\sqrt{2}) & 2(\sqrt{2}-1) & (2-\sqrt{2}) \\ \hline 0 & 0 & 0 \\ \hline \text{-}(2-\sqrt{2}) & \text{-}2(\sqrt{2}-1) & \text{-}(2-\sqrt{2}) \\ \hline \end{array}, \tag{5.34}$$

$$M_\Delta = \frac{1}{h^2} \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}. \tag{5.35}$$

El algoritmo para aplicar una máscara 3×3 a una función tabulada (es decir $F_{i,j}$ viene dada por una matriz) es un algoritmo sencillo basado en la fórmula (5.20). Sólo hay que tener un poco de cuidado en los bordes pues nos salimos de la matriz al aplicar la fórmula. En esos casos utilizamos la regla del vecino más cercano, es decir, al acceder a un valor $F_{i+k,j+l}$, si $i+k$ se sale de las dimensiones de la matriz se sustituye por i en la fórmula. De la misma forma, si $j+l$ se sale de las dimensiones de la matriz se sustituye por j en la fórmula.

Descripción algoritmo aplicación máscara a una función tabulada F_{ij} de tamaño $dim1 \times dim2$.

- Paso 1: se aplica la fórmua:

$$M * F_{i,j} = \sum_{k=-1}^{1} \sum_{l=-1}^{1} m_{k+1,l+1} F_{i+k,j+l}. \tag{5.36}$$

  al interior del rectángulo donde está definida la función dejando los bordes fuera

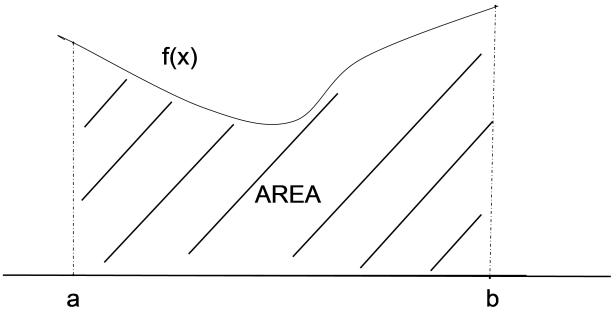
- Paso 2: se aplica la fórmula anterior a los bordes horizontales dejando las esquinas fuera.

- Paso 3: se aplica la fórmula anterior a los bordes vérticales dejando las esquinas fuera.

- Paso 4: se aplica la fórmula anterior a las esquinas.

  En los 3 últimos casos, si el punto donde hay que evaluar se sale del rectángulo donde está definida la función se toma el punto más cercano dentro del rectángulo

5.2. Integración numérica



$\int_a^b f(x)dx =$ Área encerrada por la curva y el eje x en [a,b].

Geométricamente la integral de una función positiva en un intervalo [a,b] representa el área de la región encerrada entre la función y el eje x. En este capítulo estudiaremos dos tipos de estrategias para aproximar las integrales. La primera de ellas consiste en dividir el intervalo [a,b] en subintervalos y aproximar la integral en cada subintervalo, la segunda aproxima la integral globalmente en [a,b] buscando que la integral sea exacta para polinomios hasta un cierto grado.

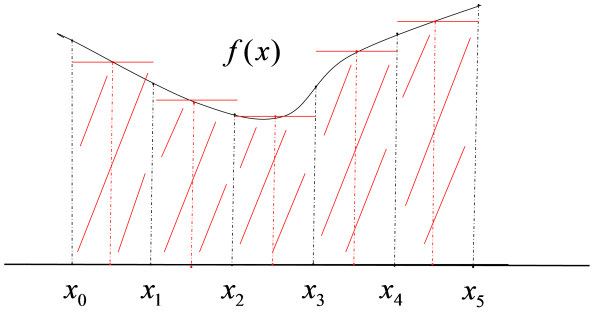
5.2.1. Fórmulas de integración numérica compuestas

Se considera una subdivisión del intervalo [a,b] dada por $a = x_0 < x_1 < .... < x_{M+1} = b$, la integral de una función $f(x)$ en [a,b] se puede expresar como

$$\int_a^b f(x)dx = \sum_{k=0}^{M} \int_{x_k}^{x_{k+1}} f(x)dx. \tag{5.37}$$

A continuación veremos diferentes formas de aproximar numéricamente cada una de las integrales dentro del sumatorio:

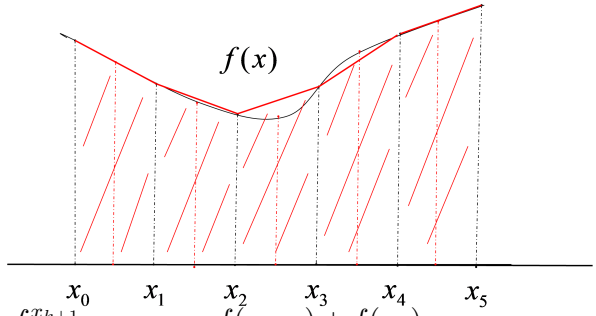
Fórmula del rectángulo



$$\int_{x_k}^{x_{k+1}} f(x)dx \approx f\left(\frac{x_k + x_{k+1}}{2}\right)(x_{k+1} - x_k)\,.$$

Esta fórmula se obtiene fácilmente aproximando la integral por el área del rectángulo de base $[x_k, x_{k+1}]$ y altura $f\left(\frac{x_k+x_{k+1}}{2}\right)$.

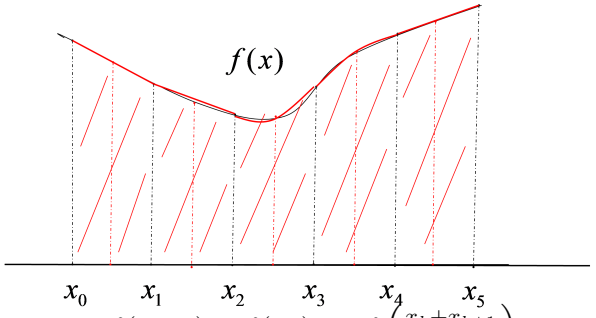
Fórmula del trapecio



$$\int_{x_k}^{x_{k+1}} f(x)dx \approx \frac{f(x_{k+1}) + f(x_k)}{2}\,(x_{k+1} - x_k)$$

Esta fórmula se obtiene fácilmente aproximando la integral por el área del trapecio encerrado por el eje x y el segmento que une los puntos $(x_k, f(x_k))$ y $(x_{k+1}, f(x_{k+1}))$.

Fórmula de Simpson



$$\int_{x_k}^{x_{k+1}} f(x)dx \approx \frac{f(x_{k+1}) + f(x_k) + 4f\left(\frac{x_k+x_{k+1}}{2}\right)}{6}\,(x_{k+1} - x_k)\,.$$

Esta fórmula se deduce sustituyendo en la integral $f(x)$ por su desarrollo en serie de Taylor hasta la segunda derivada y centrado en el punto $x_m = \frac{x_k+x_{k+1}}{2}$. Es decir:

$$\int_{x_k}^{x_{k+1}} f(x)dx \approx \int_{x_k}^{x_{k+1}} \left(f(x_m) + f'(x_m)(x - x_m) + \frac{f''(x_m)}{2}(x - x_m)^2 \right) dx = \tag{5.38}$$

$$= f(x_m)(x_{k+1} - x_k) + \frac{f''(x_m)}{3} \left(\frac{x_{k+1} - x_k}{2} \right)^3.$$

Ahora bien, teniendo en cuenta los resultados de la sección anterior sobre derivación numérica $f''(x_m)$ se puede aproximar como

$$f''(x_m) \approx \frac{f(x_{k+1}) - 2f(x_m) + f(x_k)}{\left(\frac{x_{k+1} - x_k}{2} \right)^2}. \tag{5.39}$$

Por tanto, sustituyendo este valor en la aproximación anterior obtenemos

$$\int_{x_k}^{x_{k+1}} f(x)dx \approx f(x_m)(x_{k+1} - x_k) + \frac{f(x_{k+1}) - 2f(x_m) + f(x_k)}{3} \left(\frac{x_{k+1} - x_k}{2} \right) = \tag{5.40}$$

$$= \frac{f(x_{k+1}) + f(x_k) + 4f\left(\frac{x_k + x_{k+1}}{2} \right)}{6} (x_{k+1} - x_k).$$

En general, en estas fórmulas de integración numérica, salvo indicación de lo contrario, se considera que los puntos $\{x_k\}$ son equidistantes. Para implementar estas fórmulas se usa como parámetro el número de subintervalos, N, en que se divide el intervalo inicial $[a, b]$. De tal manera que la implementación consiste simplemente en hacer el sumatorio

$$\int_a^b f(x)dx = \sum_{k=0}^{N-1} \int_{a+k\frac{b-a}{N}}^{a+(k+1)\frac{b-a}{N}} f(x)dx, \tag{5.41}$$

usando a la derecha cualquiera de las fórmulas explicadas para aproximar la integral. Se puede también implementar una versión del algoritmo que calcule de forma automática el número de intervalos, N, para alcanzar una cierta calidad en la precisión en el cálculo de la integral. Para ello se va aumentando el valor de N hasta que que el valor de la integral se estabiliza y deja de cambiar. A continuación se describe un algoritmo de este tipo para el método de Simpson

Algoritmo Simpson iterativo
variable entera N=1 (número intervalos)
variable real integral= cálculo integral de Simpson para N intervalos
variable real error=TOL+1 (error para criterio de parada)
mientras (error>TOL)
N=N*2 (se duplica el número de intervalos en cada iteración)
variable real integral2= cálculo integral de Simpson para N intervalos
error=|integral-integral2|/(|integral|+1)
integral=integral2
fin mientras
finalgoritmo

El método de Simpson en 2 variables en una región rectangular

Una integral en 2 variables en una región rectangular se puede descomponer como

$$\int_a^b \int_c^d F(x,y)dydx = \sum_{k=0}^{M-1} \sum_{n=0}^{N-1} \int_{x_k}^{x_{k+1}} \int_{y_n}^{y_{n+1}} F(x,y)dydx, \tag{5.42}$$

en cada rectángulo $[x_k, x_{k+1}]\times[y_n, y_{n+1}]$ aplicamos Simpson en 2 fases de la siguiente manera:

$$\int_{x_k}^{x_{k+1}} \underbrace{\int_{y_n}^{y_{n+1}} F(x,y)dy}_{f(x)} dx \simeq \frac{f(x_k)+4f(x_m)+f(x_{k+1})}{6}(x_{k+1}-x_k), \tag{5.43}$$

donde

$$f(x) = \int_{y_n}^{y_{n+1}} F(x,y)dy \simeq \frac{F(x,y_n)+4F(x,y_m)+F(x,y_{n+1})}{6}(y_{n+1}-y_n). \tag{5.44}$$

Esto se puede hacer igualmente para los métodos del rectángulo y del trapecio.

5.2.2. Métodos de cuadratura de Gauss

Sea $f(x)$ una función definida en un intervalo $[a,b]$, vamos a aproximar el valor de la integral de $f(x)$ en $[a,b]$ utilizando la evaluación de $f(x)$ en ciertos puntos de $[a,b]$. Es decir, una fórmula de integración por cuadratura de Gauss se puede escribir como

$$\int_a^b f(x)dx \approx \sum_{k=1}^{N} w_k f(x_k), \tag{5.45}$$

donde x_k representa los puntos de evaluación de $f(x)$ y w_k el peso de cada punto de evaluación.

Definición 2 *Una fórmula de integración numérica se denomina exacta de orden M si, para cualquier polinomio $P(x)$ de grado menor o igual que M, la fórmula es exacta. Es decir*

$$\int_a^b P(x)dx = \sum_{k=1}^{N} w_k P(x_k). \tag{5.46}$$

Definición 3 *Se denominan polinomios de Legendre $L_n(x)$ a la familia de polinomios dada por $L_0(x) = 1$, $L_1(x) = x$, y para $n = 2, 3, ...$*

$$nL_n(x) = (2n-1)xL_{n-1}(x) - (n-1)L_{n-2}(x). \tag{5.47}$$

Teorema 17 *Sean $\{\tilde{x}_k\}_{k=1,..,N}$ los ceros del polinomio de Legendre $L_N(x)$. Si definimos*

$$\tilde{w}_k = \int_{-1}^{1} \frac{\Pi_{i\neq k}(x - \tilde{x}_i)}{\Pi_{i\neq k}(\tilde{x}_k - \tilde{x}_i)} dx, \tag{5.48}$$

entonces la fórmula de integración numérica generada por los puntos $\tilde{x}_k$ y los pesos $\tilde{w}_k$ es exacta hasta el orden $2N-1$ para el intervalo $[-1, 1]$.

Demostración [Hu].

Ejemplo 9 *A continuación se presentan algunos valores de raíces $\tilde{x}_k$ y coeficientes $\tilde{w}_k$ en función del grado del polinomio $L_N(x)$:*

N	$\tilde{x}_k$	$\tilde{w}_k$
2	0.5773502692	1.
	-0.5773502692	1
3	0.7745966692	0.5555555556
	0.	0.8888888889
	$-$ 0.7745966692	0.5555555556
4	0.8611363116	0.3478548451
	0.3399810436	0.6251451549
	-0.3399810436	0.6251451549
	$-$ 0.8611363116	0.3478548451

La tabla de puntos y pesos anteriores está referenciado al intervalo $[-1, 1]$, para el caso general de un intervalo $[a, b]$ hay que hacer un cambio de variables para llevar el intervalo $[a, b]$ a $[-1, 1]$. Es decir, aplicando el cambio de variables

$$z = \frac{(b-a)\,x + b + a}{2}, \tag{5.49}$$

a la integral se obtiene:

$$\int_a^b f(z)\,dz = \int_{-1}^{1} \frac{b-a}{2} f\left(\frac{(b-a)\,x+b+a}{2}\right) dx \simeq \sum_{k=1}^{N} \tilde{w}_k \frac{b-a}{2} f\left(\frac{(b-a)\,\tilde{x}_k+b+a}{2}\right). \tag{5.50}$$

Cuando el intervalo $[a,b]$ es infinito, es decir, $a=-\infty$ o $b=\infty$, hay que emplear otros métodos para aproximar las integrales. En el caso $[a,b]=(-\infty,\infty)$, se utilizan los ceros de los denominados polinomios de Hermite, definidos como $H_0(x)=1$, $H_1(x)=2x$, y

$$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x), \tag{5.51}$$

para $n \geq 2$. En este caso, la fórmula de integración numérica aproxima la integral de la siguiente forma:

$$\int_{-\infty}^{\infty} f(x)e^{-x^2}dx \approx \sum_{k=0}^{N} w_k f(x_k). \tag{5.52}$$

Teorema 18 *Si $\tilde{x}_k$ son los ceros del polinomio de Hermite y definimos*

$$\tilde{w}_k = \int_{-\infty}^{\infty} \frac{\Pi_{i\neq k}(x-\tilde{x}_i)}{\Pi_{i\neq k}(x_k-\tilde{x}_i)} e^{-x^2}dx, \tag{5.53}$$

entonces la fórmula de integración numérica generada por los puntos $\tilde{x}_k$ y los pesos $\tilde{w}_k$ es exacta hasta el orden $2N-1$ para el intervalo $(-\infty,\infty)$.

Demostración [Hu].

Ejemplo 10 *A continuación se presentan algunos valores de raíces $\tilde{x}_k$ y coeficientes $\tilde{w}_k$ en función del grado del polinomio $H_N(x)$:*

N	$\tilde{x}_k$	$\tilde{w}_k$
1	0.	1.772453851
2	−0.707106781	0.886226925
	0.707106781	0.886226925
3	−1.224744871	0.295408975
	0.	1.181635900
	1.224744871	0.295408975

Para el intervalo $(0,\infty)$, se utilizan los polinomios de Laguerre $\hat{L}_n(x)$, definidos por $\hat{L}_0(x)=1$, $\hat{L}_1(x)=1-x$, y

$$\hat{L}_n(x) = (2n-1-x)\hat{L}_{n-1}(x) - (n-1)^2\hat{L}_{n-2}(x), \tag{5.54}$$

para $n \geq 2$. En este caso, la fórmula de integración numérica aproxima:

$$\int_0^{\infty} f(x)e^{-x}dx \approx \sum_{k=0}^{N} w_k f(x_k). \tag{5.55}$$

Teorema 19 *Si $\tilde{x}_k$ son los ceros del polinomio de Laguerre y definimos*

$$\tilde{w}_k = \int_0^\infty \frac{\Pi_{i\neq k}(x-\tilde{x}_i)}{\Pi_{i\neq k}(x_k-\tilde{x}_i)} e^{-x} dx, \tag{5.56}$$

entonces la fórmula de integración numérica generada por los puntos $\tilde{x}_k$ y los pesos $\tilde{w}_k$ es exacta hasta el orden $2N-1$ para el intervalo $(0,\infty)$.

Demostración [Hu].

Ejemplo 11 *A continuación se presentan algunos valores de raíces $\tilde{x}_k$ y coeficientes $\tilde{w}_k$ en función del grado del polinomio $\hat{L}_N(x)$:*

N	$\tilde{x}_k$	$\tilde{w}_k$
1	1.	1.
2	0.585787	0.853553
	3.414214	0.146447
3	0.415775	0.711093
	2.29428	0.278518
	6.28995	0.010389

Métodos de cuadratura de Gauss en 2 variables

Para aplicar los métodos de cuadratura a 2 variables se aplica dos veces de forma consecutiva el método de una variable. Es decir:

$$\int_a^b \underbrace{\int_c^d F(x,y)dy}_{f(x)} dx \approx \sum_{k=1}^{N} w_k f(x_k) \approx \sum_{k=1}^{N} w_k \sum_{k'=1}^{N'} w'_{k'} F(x_k, y_{k'}) = \tag{5.57}$$

$$= \sum_{k=1}^{N}\sum_{k'=1}^{N'} \underbrace{W_{k,k'}}_{w_k w'_{k'}} F(x_k, y_{k'}), \tag{5.58}$$

por tanto, si $\{x_k, w_k\}_{k=1,..,N}$ y $\{y_{k'}, w'_{k'}\}_{k'=1,..,N'}$ son los puntos y pesos de las fórmulas de cuadratura en los intervalos $[a,b]$ y $[c,d]$ respectivamente, entonces los puntos $\{(x_k, y_{k'})\}$ con los pesos $W_{k,k'} = w_k w'_{k'}$ son los puntos y pesos de la fórmula de cuadratura para la región $[a,b]\times[c,d]$.

5.3. Problemas resueltos

Problema 43 *Dada una función $f(x)$ donde x se mueve en el rango de valores $[2^5, 2^8]$, si aproximamos la derivada, en una aritmética de 32 bits, usando la fórmula*

$$f'(x) = \frac{f(x+h)-f(x)}{h},$$

cual es el valor de h que podemos tomar para todo el intervalo.

Solución: el valor de h para un intervalo viene dado por

$$h = max\{|a|, |b|\}\sqrt{u},$$

donde $u = \frac{1}{2^t}$ es la unidad de redondeo de la aritmética. En en una aritmética de 32 bits $t = 23$ y por tanto

$$h = 2^8\sqrt{\frac{1}{2^{23}}} = 2^{8-\frac{23}{2}} \approx 2^{-4}.$$

Problema 44 *Tomando $h = 0.25$ y usando la fórmula de aproximación de las derivadas de orden $O(h)$ y $O(h^2)$ en dimensión 1, aproximar las derivadas parciales de la función $F(x, y) = x^4y^5$ en el punto $(x, y) = (1, 1)$*

Solución: usando las fórmulas para el cálculo de la derivada primera se obtiene:

$$\begin{aligned}
O(h) \quad &\to f'(x) \approx \frac{f(x+h) - f(x)}{h}, \\
O(h) \quad &\to \quad \frac{\partial F(1,1)}{\partial x} \approx \frac{(1+h)^4 1^5 - 1^4 1^5}{h} = 5.765625, \\
O(h) \quad &\to \quad \frac{\partial F(1,1)}{\partial y} \approx \frac{1^4(1+h)^5 - 1^4 1^5}{h} = 8.207031.
\end{aligned}$$

$$\begin{aligned}
O(h^2) \quad &\to f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}, \\
O(h^2) \quad &\to \quad \frac{\partial F(1,1)}{\partial x} \approx \frac{(1+h)^4 1^5 - (1-h)^4 1^5}{2h} = 4.25, \\
O(h^2) \quad &\to \quad \frac{\partial F(1,1)}{\partial y} \approx \frac{1^4(1+h)^5 - 1^4(1-h)^5}{2h} = 5.6289.
\end{aligned}$$

Problema 45 *Dada la función $f(x) = 2x^3$ aproximar la derivada f'(1) utilizando la evaluación de la función en los puntos x=1 y x=1.25.*

Solución: la distancia entre los puntos es $h = 1.25 - 1 = 0.25$, como se usa el propio punto donde se va a calcular la derivada hay que usar la fórmula

$$f'(1) = \frac{f(1+h) - f(1)}{h} = \frac{2(1+h)^3 - 2(1)^3}{h} = 7.625.$$

Problema 46 *Dada la función $f(x) = 2x^3$ aproximar la derivada f'(1) utilizando la evaluación de la función en los puntos x=0.75 y x=1.25.*

Solución: la distancia entre los puntos es $h = 1.25 - 1 = 0.25$, como no se usa el propio punto donde se va a calcular la derivada hay que usar la fórmula

$$f'(1) = \frac{f(1+h) - f(1-h)}{2h} = \frac{2(1+h)^3 - 2(1-h)^3}{2h} = 6.125$$

Problema 47 *Dada la función $f(x) = 2x^3$ aproximar la derivada segunda f"(1) utilizando la evaluación de la función en los puntos x=0.75, x=1 y x=1.25.*

Solución: la distancia entre los puntos es $h = 1.25 - 1 = 0.25$, aplicando la fórmula de la derivada segunda se obtiene

$$f'(1) = \frac{f(1+h) - 2f(1) + f(1-h)}{h^2} = \frac{2(1+h)^3 - 2 \cdot 2(1)^3 + 2(1-h)^3}{h^2} = 12.$$

Problema 48 *Calcular una aproximación de la derivada primera y segunda de una función $f(x)$ en $x = 0$, teniendo en cuenta que $f(0) = 1$, $f(1) = 0$, $f(4) = 9$.*

Solución: usando la fórmula de la derivada se obtiene:

$$f'(x_i) \approx \frac{(x_i - x_l)\frac{f(x_r)-f(x_i)}{x_r - x_i} + (x_r - x_i)\frac{f(x_i)-f(x_l)}{x_i - x_l}}{x_r - x_l} = \frac{-1\frac{f(4)-f(0)}{4-0} + 4\frac{f(0)-f(1)}{0-1}}{4-1} = -2,$$

$$f''(x_i) \approx 2\frac{\frac{f(x_r)-f(x_i)}{(x_r - x_i)} - \frac{f(x_i)-f(x_l)}{(x_i - x_l)}}{x_r - x_l} = 2\frac{\frac{9-1}{(4-0)} - \frac{1-0}{(0-1)}}{4-1} = 2.$$

Problema 49 *Calcular una aproximación de la derivada tercera $f'''(x_i)$ de una función $f(x)$ en un punto x_i, utilizando $f(x_i)$, $f(x_i + h)$, $f(x_i - h)$, $f(x_i - 2h)$*

Solución: utilizaremos los desarrollos de Taylor en los puntos que vamos a usar:

$$f(x_i + h) = f(x_i) + hf'(x_i) + \frac{h^2}{2}f''(x_i) + \frac{h^3}{6}f'''(x_i) + O(h^4),$$

$$f(x_i - h) = f(x_i) - hf'(x_i) + \frac{h^2}{2}f''(x_i) - \frac{h^3}{6}f'''(x_i) + O(h^4),$$

$$f(x_i - 2h) = f(x_i) - 2hf'(x_i) + 2h^2 f''(x_i) - \frac{4h^3}{3}f'''(x_i) + O(h^4),$$

combinando estos desarrollos con unos pesos dados por a, b y c obtenemos

$$af(x_i + h) + bf(x_i - h) + cf(x_i - 2h) = (a + b + c)f(x_i) + (a - b - 2c)hf'(x_i) +$$

$$\left(\frac{a}{2} + \frac{b}{2} + 2c\right)h^2 f''(x_i) + \left(\frac{a}{6} - \frac{b}{6} - \frac{4c}{3}\right)h^3 f'''(x_i) + O(h^4),$$

por tanto las condiciones que tienen que darse para aproximar $f'''(x_i)$ son

$$\begin{aligned} a - b - 2c &= 0, \\ \tfrac{a}{2} + \tfrac{b}{2} + 2c &= 0, \\ \tfrac{a}{6} - \tfrac{b}{6} - \tfrac{4c}{3} &= 1, \end{aligned}$$

resolviendo este sistema obtenemos $a = 1, b = 3, c = -1$, y finalmente despejando obtenemos

$$f'''(x_i) = \frac{f(x_i + h) + 3f(x_i - h) - f(x_i - 2h) - 3f(x_i)}{h^3} + O(h).$$

Problema 50 *Se considera una función tabulada dada por la matriz*

$$F = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \end{pmatrix},$$

y la máscara

$$M = \begin{array}{|c|c|c|} \hline 0 & 0.25 & 0 \\ \hline 0.25 & -1 & 0.25 \\ \hline 0 & 0.25 & 0 \\ \hline \end{array}.$$

Calcular $M * F_{0,0}$, $M * F_{1,0}$, *y* $M * F_{1,1}$.

Solución: en los casos en que al aplicar la máscara nos salimos de la matriz extendemos la matriz original por el criterio del vecino más cercano. Por tanto

$$M * F_{0,0} = \begin{array}{|c|c|c|} \hline 0 & 0.25 & 0 \\ \hline 0.25 & -1 & 0.25 \\ \hline 0 & 0.25 & 0 \\ \hline \end{array} \cdot \begin{array}{|c|c|c|} \hline 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ \hline 1 & 1 & 2 \\ \hline \end{array} = 0.5,$$

la multiplicación de las dos matrices 3×3 se interpreta como la multiplicación y suma de sus coeficientes. De la misma forma, para los otros casos se obtiene

$$M * F_{1,0} = \begin{array}{|c|c|c|} \hline 0 & 0.25 & 0 \\ \hline 0.25 & -1 & 0.25 \\ \hline 0 & 0.25 & 0 \\ \hline \end{array} \cdot \begin{array}{|c|c|c|} \hline 0 & 0 & 1 \\ \hline 1 & 1 & 2 \\ \hline 2 & 2 & 3 \\ \hline \end{array} = 0.25,$$

$$M * F_{1,1} = \begin{array}{|c|c|c|} \hline 0 & 0.25 & 0 \\ \hline 0.25 & -1 & 0.25 \\ \hline 0 & 0.25 & 0 \\ \hline \end{array} \cdot \begin{array}{|c|c|c|} \hline 0 & 1 & 2 \\ \hline 1 & 2 & 3 \\ \hline 2 & 3 & 4 \\ \hline \end{array} = 0.$$

Problema 51 *Usando las máscaras*

$$M_x = \frac{1}{4h} \begin{array}{|c|c|c|} \hline -(2-\sqrt{2}) & 0 & (2-\sqrt{2}) \\ \hline -2(\sqrt{2}-1) & 0 & 2(\sqrt{2}-1) \\ \hline -(2-\sqrt{2}) & 0 & (2-\sqrt{2}) \\ \hline \end{array},$$

$$M_y = \frac{1}{4h} \begin{array}{|c|c|c|} \hline (2-\sqrt{2}) & 2(\sqrt{2}-1) & (2-\sqrt{2}) \\ \hline 0 & 0 & 0 \\ \hline -(2-\sqrt{2}) & -2(\sqrt{2}-1) & -(2-\sqrt{2}) \\ \hline \end{array},$$

calcular numéricamente la matriz gradiente en el punto $(1,1)$ *de la función*

$$f(x,y) = \begin{cases} x^2 + y^2 - 1, \\ x - y. \end{cases}$$

(utilizar $h = 0.1$*) y compararla con el resultado de la matriz gradiente calculada analíticamente.*

Solución: el cálculo analítico de la matriz gradiente nos da

$$\nabla f(x,y) = \begin{pmatrix} 2x & 2y \\ 1 & -1 \end{pmatrix} \to \nabla f(1,1) = \begin{pmatrix} 2 & 2 \\ 1 & -1 \end{pmatrix}.$$

Numéricamente, llamamos $f_1(x,y) = x^2 + y^2 - 1$ y $f_2(x,y) = x - y$. Construimos los valores discretos de la funciones y la expresamos en forma de matriz

$$F_1 = \begin{array}{|c|c|c|} \hline f_1(1-h,1+h) & f_1(1,1+h) & f_1(1+h,1+h) \\ \hline f_1(1-h,1) & f_1(1,1) & f_1(1+h,1) \\ \hline f_1(1-h,1-h) & f_1(1,1-h) & f_1(1+h,1-h) \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 2.02 & 2.21 & 2.42 \\ \hline 1.81 & 2 & 2.21 \\ \hline 1.62 & 1.81 & 2.02 \\ \hline \end{array},$$

$$F_2 = \begin{array}{|c|c|c|} \hline f_2(1-h,1+h) & f_2(1,1+h) & f_2(1+h,1+h) \\ \hline f_2(1-h,1) & f_2(1,1) & f_2(1+h,1) \\ \hline f_2(1-h,1-h) & f_2(1,1-h) & f_2(1+h,1-h) \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline -0.2 & -0.1 & 0 \\ \hline -0.1 & 0 & 0.1 \\ \hline 0 & 0.1 & 0.2 \\ \hline \end{array},$$

por tanto numéricamente usando las máscaras tenemos

$$\nabla f(1,1) \approx \begin{pmatrix} M_x * F_1 & M_y * F_1 \\ M_x * F_2 & M_y * F_2 \end{pmatrix},$$

cada uno de los coeficientes de la matriz se obtiene multiplicando y sumando las matrices asociadas, es decir por ejemplo

$$M_x * F_1 = \frac{1}{4h} \begin{array}{|c|c|c|} \hline -(2-\sqrt{2}) & 0 & (2-\sqrt{2}) \\ \hline -2(\sqrt{2}-1) & 0 & 2(\sqrt{2}-1) \\ \hline -(2-\sqrt{2}) & 0 & (2-\sqrt{2}) \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 2.02 & 2.21 & 2.42 \\ \hline 1.81 & 2 & 2.21 \\ \hline 1.62 & 1.81 & 2.02 \\ \hline \end{array} =$$

$$= \frac{1}{0.4}(-(2-\sqrt{2})2.02 + (2-\sqrt{2})2.42 - 2(\sqrt{2}-1)1.81 +$$
$$+2(\sqrt{2}-1)2.21 - (2-\sqrt{2})1.62 + (2-\sqrt{2})2.02) = 2,$$

haciendo lo mismo para el resto de matrices obtenemos

$$\nabla f(1,1) \approx \begin{pmatrix} M_x * F_1 & M_y * F_1 \\ M_x * F_2 & M_y * F_2 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 1 & -1 \end{pmatrix}.$$

Con lo cual, en este caso la matriz gradiente calculada numéricamente coincide con la calculada analíticamente.

Problema 52 *Calcular una aproximación del laplaciano en el punto central de una función $F_{i,j}$ tabulada que tiene la forma*

$$F = \begin{array}{|c|c|c|} \hline \frac{1}{2} & \frac{1}{4} & \frac{1}{2} \\ \hline \frac{1}{4} & 0 & \frac{1}{4} \\ \hline \frac{1}{2} & \frac{1}{4} & \frac{1}{2} \\ \hline \end{array},$$

y tal que la distancia entre los puntos es $h = \frac{1}{2}$.

Solución: la máscara del laplaciano que hemos estudiado es

$$M_\Delta = \frac{1}{h^2}\begin{array}{|c|c|c|}\hline 0&1&0\\\hline 1&-4&1\\\hline 0&1&0\\\hline\end{array} = 4\begin{array}{|c|c|c|}\hline 0&1&0\\\hline 1&-4&1\\\hline 0&1&0\\\hline\end{array},$$

por tanto la aproximación del laplaciano viene dada por

$$M_\Delta * F_{1,1} = 4\begin{array}{|c|c|c|}\hline 0&1&0\\\hline 1&-4&1\\\hline 0&1&0\\\hline\end{array}\cdot\begin{array}{|c|c|c|}\hline \frac{1}{2}&\frac{1}{4}&\frac{1}{2}\\\hline \frac{1}{4}&0&\frac{1}{4}\\\hline \frac{1}{2}&\frac{1}{4}&\frac{1}{2}\\\hline\end{array} = 4(\frac{1}{4}+\frac{1}{4}+\frac{1}{4}+\frac{1}{4}-4\cdot 0) = 4.$$

Problema 53 *Aproximar la integral*

$$\int_3^5 2x^3 dx,$$

usando la fórmula del rectángulo y dividiendo el intervalo $[3,5]$ *en 4 intervalos*

Solución: los intervalos serían $[3, 3.5]$, $[3.5, 4]$, $[4, 4.5]$ y $[4.5, 5]$. Por tanto usando la fórmula del rectángulo la integral se calcula haciendo

$$\int_3^5 2x^3 dx \approx 2(\frac{3+3.5}{2})^3\cdot 0.5+2(\frac{3.5+4}{2})^3\cdot 0.5+2(\frac{4+4.5}{2})^3\cdot 0.5+2(\frac{4.5+5}{2})^3\cdot 0.5 = 271.$$

Problema 54 *Aproximar la integral*

$$\int_3^5 2x^3 dx,$$

usando la fórmula del trapecio y dividiendo el intervalo $[3,5]$ *en 2 intervalos*

Solución: los intervalos serían $[3, 4]$, y $[4, 5]$. Por tanto usando la fórmula del trapecio la integral se calcula haciendo

$$\int_3^5 2x^3 dx \approx \frac{2(3)^3+2(4)^3}{2}1+\frac{2(4)^3+2(5)^3}{2}1 = 280.$$

Problema 55 *Aproximar la integral*

$$\int_3^5 2x^3 dx,$$

usando la fórmula del Simpson y dividiendo el intervalo $[3,5]$ *en 2 intervalos*

Solución: los intervalos serían $[3, 4]$, y $[4, 5]$. Por tanto usando la fórmula de Simpson la integral se calcula haciendo

$$\int_3^5 2x^3 dx \approx \frac{2(3)^3+4\cdot 2(3.5)^3+2(4)^3}{6}1+\frac{2(4)^3+4\cdot 2(4.5)^3+2(5)^3}{6}1 = 272.$$

Problema 56 *Aproximar el valor de la siguiente integral, utilizando la cuadratura de Gauss para $N = 2$ y $N = 3$*

$$\int_{-1}^{1} \left(x^3 - x^4\right) dx.$$

Cual es el valor exacto de la integral?

Solución:

$\int_{-1}^{1} \left(x^3 - x^4\right) dx \simeq \sum_{k=0}^{N} w_k P\left(x_k\right)$,

$P\left(x\right) = x^3 - x^4$.

1. $N = 2$.

$$\sum_{k=1}^{2} w_k P\left(x_k\right) = 1 \cdot P(0.5773502692) + 1 \cdot P\left(-0.5773502692\right) = -0.22222.$$

2. $N = 3$.

$$\sum_{k=1}^{3} w_k P\left(x_k\right) = 0.\hat{5} \cdot P\left(0.774596\right) + 0.\hat{8} \cdot P\left(0\right) + 0.\hat{5} \cdot P\left(-0.77459\right) = -0.4.$$

El valor exacto de la integral es $\int_{-1}^{1} \left(x^3 - x^4\right) dx = -\frac{2}{5} = -0.4$, que coincide con el valor del segundo caso. La fórmula de integración numérica es exacta hasta el orden $2N - 1$, que en el segundo caso es equivalente a 5, con lo que ya se sabía que el valor obtenido sería exacto.

Problema 57 *Deducir la fórmula de integración numérica en el intervalo $[-1, 1]$ utilizando un sólo punto de interpolación, y de tal manera que sea exacta para polinomios de hasta grado 1*

Solución: la fórmula que usa un único punto se puede expresar como

$$\int_{-1}^{1} f\left(x\right) dx \simeq w_0 \cdot f\left(x_0\right).$$

Vamos a imponer que se exacta para los polinomios $f(x) = 1$ y $f(x) = x$

$$\int_{-1}^{1} 1 dx = 2 = w_0 \cdot f\left(x_0\right) = w_0 \quad \rightarrow \quad w_0 = 2,$$

$$\int_{-1}^{1} x dx = 0 = w_0 \cdot f(x_0) = w_0 \cdot x_0 \quad \rightarrow \quad x_0 = 0.$$

Por lo tanto, la fórmula de integración numérica es:

$$\int_{-1}^{1} f(x)\, dx \simeq 2 \cdot f(0),$$

Problema 58 *A partir de los ceros y de los pesos asociados a la cuadratura de Gauss en* $[0,1]$*, y dado un intervalo* $[a,b]$ *cualquiera, encontrar los puntos* x_k*, y los pesos* w_k *que hacen exacta hasta el orden* $2N-1$ *una fórmula de integración numérica sobre el intervalo* $[a,b]$

Solución: Para encontrar los puntos $\hat{x}_k$, y los pesos $\hat{w}_k$, hay que hacer un cambio de variable en la integral:

$$\int_a^b f(x)\, dx \simeq \sum_{k=1}^{N} \hat{w}_k f(\hat{x}_k)$$

Hacemos el siguiente cambio de variable:

$$\begin{array}{l} x = \frac{(b-a)t+(b+a)}{2} \\ dx = \frac{b-a}{2} dt, \end{array}$$

este cambio transforma el intervalo $[-1,1]$ en el intervalo $[a,b]$ y por tanto

$$\int_a^b f(x)\, dx = \int_{-1}^{1} f\left(\frac{(b-a)\,t+b+a}{2}\right) \frac{b-a}{2} dt,$$

$$\int_a^b f(x)\, dx \simeq \sum_{k=1}^{N} \tilde{w}_k \frac{b-a}{2} f\left(\frac{(b-a)\,\tilde{x}_k+b+a}{2}\right),$$

de donde se deduce que los cambios a realizar son de la forma

$$\begin{array}{l} \hat{x}_k = \frac{(b-a)\tilde{x}_k+(b+a)}{2}, \\ \hat{w}_k = \frac{(b-a)}{2}\tilde{w}_k. \end{array}$$

Problema 59 *Utilizar el resultado del problema anterior para calcular de forma exacta la siguiente integral*

$$\int_0^1 \left(x^2 - x^3\right) dx.$$

Solución: el resultado de la integral calculada de forma analítica, da el siguiente resultado:

$\int_0^1 \left(x^2 - x^3\right) dx = \frac{1}{12} = 8.\hat{3} \times 10^{-2}$.

Aplicando el método de integración numérica:

$f(x) = x^2 - x^3$,

$$\int_0^1 \left(x^2 - x^3\right) dx = \sum_{k=1}^{3} w_k f(x_k) = w_1 f(x_1) + w_2 f(x_2) + w_3 f(x_3) =$$

$$= \left(\tfrac{1-0}{2}\right)\left(\tilde{w}_0 f\left(\tfrac{\tilde{x}_0+1}{2}\right) + \tilde{w}_1 f\left(\tfrac{\tilde{x}_1+1}{2}\right) + \tilde{w}_2 f\left(\tfrac{\tilde{x}_2+1}{2}\right)\right) = \tfrac{1}{2}(0.\hat{5} \cdot f\left(\tfrac{0.7745966692+1}{2}\right) +$$

$$+0.\hat{8} \cdot f\left(\tfrac{0+1}{2}\right) + 0.\hat{5} \cdot f\left(\tfrac{-0.7745966692+1}{2}\right)) = 8.\hat{3} \times 10^{-2}.$$

Problema 60 *Calcular de forma exacta la integral*

$$\int_{-\infty}^{\infty} \left(x^3 - x^2\right) e^{-x^2} dx,$$

utilizando cuadratura de Gauss.

Solución: de forma analítica la integral da como resultado:

$\int_{-\infty}^{\infty} \left(x^3 - x^2\right) e^{-x^2} dx = -\frac{1}{2}\sqrt{\pi} = -0.88623$.

Utilizando el método de integración numérica:

$f(x) = x^3 - x^2$,

$\int_{-\infty}^{\infty} \left(x^3 - x^2\right) e^{-x^2} dx = \sum_{k=1}^{2} w_k f(x_k) = w_1 f(x_1) + w_2 f(x2) = 0.8862269255 \cdot f(-0.707106781) + 0.8862269255 \cdot f(0.707106781) = -0.88623$

Problema 61 *Aproximar, utilizando dos puntos de aproximación, el valor de la integral:*

$$\int_{-\infty}^{\infty} \frac{1}{1+x^2} dx$$

Solución: la solución analítica es $\int_{-\infty}^{\infty} \frac{1}{1+x^2} dx = \pi$. A continuación usamos la fórmula de cuadratura con dos puntos :

$$\int_{-\infty}^{\infty} \frac{1}{1+x^2} dx = \int_{-\infty}^{\infty} \underbrace{\frac{e^{x^2}}{1+x^2}}_{f(x)} e^{-x^2} dx \simeq$$

$$\simeq 0.886226 \cdot f(-0.70710) + 0.886226 \cdot f(0.70710) = 1.9482$$

Problema 62 *Calcular de forma exacta la integral*

$$\int_0^\infty \left(x^3 - x^2\right) e^{-x} dx$$

utilizando cuadratura de Gauss.

Solución: como el grado del polinomio es 3 es suficiente utilizar la fórmula de 2 puntos

$$\int_0^\infty \left(x^3 - x^2\right) e^{-x} dx = 0.8535533903 \cdot f\left(0.585786438\right) +$$
$$+0.1464466093 \cdot f\left(3.414213562\right) = 4.0$$

Problema 63 *Calcular una fórmula de aproximación numérica de la integral siguiente*

$$\int_a^\infty f(x) e^{-x} dx,$$

donde a es un número real cualquiera a partir de la fórmula para la integral en $[0, \infty)$.

Solución: para calcular esta integral realizamos el cambio de variable $t = x - a$

$$\int_a^\infty f(x) e^{-x} dx = \int_0^\infty f(t+a) e^{-t-a} dt = e^{-a} \int_a^\infty f(t+a) e^{-t} dt$$

por tanto usando los puntos, $\tilde{x}_k$, y pesos, $\tilde{w}_k$, para la integral en $[0, \infty)$ se obtiene

$$\int_a^\infty f(x) e^{-x} dx \approx \sum_{k=0}^{N} e^{-a} \tilde{w}_k f\left(\tilde{x}_k + a\right)$$

Problema 64 *Deducir la fórmula de integración numérica sobre un rectángulo* $[a,b] \times [c,d]$ *resultante de aplicar la integración numérica en una variable en los intervalos* $[a,b]$, *y* $[c,d]$.

Solución:

$$\int_a^b \int_c^d F\left(x,y\right) dy dx = \int_a^b \sum_{j=1}^{N} \tilde{w}'_j F\left(x, \tilde{y}_k\right) dx = \sum_{k=1}^{N} \sum_{j=1}^{N} \tilde{w}_k \tilde{w}'_j F\left(\tilde{x}_k, \tilde{y}_k\right),$$

por tanto, teniendo en cuenta los resultados obtenidos al integrar en una variable tenemos que :

$$\begin{aligned} \tilde{x}_k &= \tfrac{(b-a)x_k + (b+a)}{2}, \\ \tilde{w}_k &= \tfrac{(b-a)}{2} w_k, \\ \tilde{y}_k &= \tfrac{(d-c)y_k + (d+c)}{2}, \\ \tilde{w}'_j &= \tfrac{(d-c)}{2} w_k, \end{aligned}$$

donde w_k son los pesos al integrar en una variable en el intervalo $[-1, 1]$.

5.4. Aplicación en Epidemiología

En este apartado vamos a ver una aplicación de la integración numérica a la evaluación de la "renewal equation" que se puede traducir al español como la ecuación de reemplazo generacional. Esta ecuación determina como se van propagando los contagios de una enfermedad. Los elementos que intervienen en esta ecuación son :

1. La tasa de reproducción $R(t)$ que nos indica cuantas personas contagia, en media, cada infectado. Esta tasa de reproducción depende de la capacidad del virus para propagarse, lo cual puede cambiar con la aparición de nuevas variantes y de las medidas de distanciamiento social establecidas por los gobiernos para evitar la propagación. Cuando $R(t)$ es mayor que 1, la epidemia se encuentra en expansión y el número de contagios diarios crece. Cuando $R(t)$ es menor que 1, el número de contagios diarios tiende a decrecer. Las medidas de distanciamiento social tienen por objetivo que $R(t)$ descienda por debajo de 1.

2. La capacidad de contagiar de una persona a partir del momento que se contagia. La capacidad para contagiar a otros de una persona va variando a lo largo del tiempo desde el momento que se contagia. El denominado "tiempo de generación" que se denota por $\Phi(s)$ determina la probabilidad de que una persona contagie a otra s días después de haberse contagiado, típicamente, $\Phi(s)$ es pequeño los primeros días, después va aumentando hasta llegar al día de máxima capacidad para contagiar y después va bajando hasta llegar a 0 cuando ya la persona ya no puede contagiar a otras. Llamaremos $Max(s)$ al máximo número de días que una persona puede contagiar a otras.

A continuación vamos a deducir la "renewal equation". Para ello llamaremos $i(t)$ al número de personas que se contagian el día t. Teniendo en cuenta la tasa de reprodución $R(t)$, el número total de nuevos contagios que se generan a partir de los contagiados el día t es $i(t)R(t)$. Estos contagios se van distribuyendo en el tiempo en base a la distribución $\Phi(s)$, es decir, en el día $t+s$, el número de nuevos contagios generados a partir de los contagios del día t es $i(t)R(t)\Phi(s)$. De la misma forma podemos proceder hacia atrás y decir que el número de contagios que aportan el día t los contagiados s días antes es $i(t-s)R(t-s)\Phi(s)$. Por tanto, podemos calcular $i(t)$ a partir de los contagiados los días anteriores utilizando la fórmula

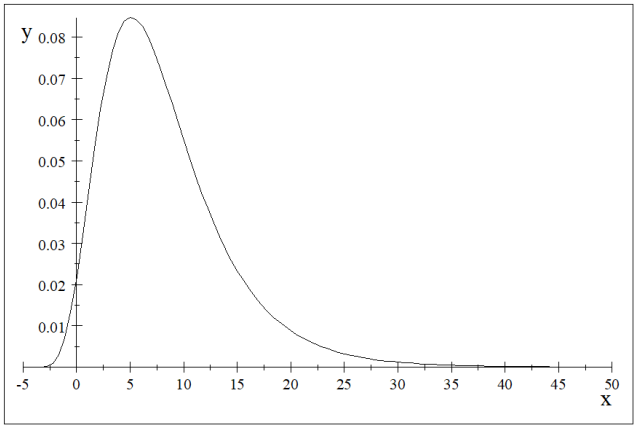
$$i(t) = \sum_{s=1}^{Max(s)} i(t-s)R(t-s)\Phi(s) \tag{5.59}$$

esta es la denominada "renewal equation" en su versión discreta. La versión continua de esta ecuación se obtiene transformado la suma en una integral haciendo

$$i(t) = \int_0^{Max(s)} i(t-s)R(t-s)\Phi(s)ds. \tag{5.60}$$

En la práctica, el valor real del número de contagios diarios $i(t)$ no es un dato observable en el sentido de que, en general, es difícil (o imposible) establecer el día que se contagió una persona. El dato que comunican los gobiernos es el número de test positivos confirmados en el día t que es bastante distinto al número de contagiados ese día, por ello, los datos que comunican los gobiernos llevan un retraso importante respecto a los contagios reales debido al tiempo de incubación del virus y al tiempo para realizar y contabilizar los tests. Es decir el dato oficial de un día nos está dando información de como se estaba propagando el virus hace más de 10 días, y por ello, cada vez que se implementa una nueva medida de distanciamiento social hay que esperar más de 10 días para observar su impacto en el número de casos diarios registrados. Para adaptar mejor la "renewal equation" al dato observable, $i(t)$, que comunican los gobiernos, la distribución $\Phi(s)$ que representa la probabilidad del tiempo que pasa entre contagios se sustituye por la probabilidad del tiempo que pasa entre que una persona presenta síntomas (caso primario) y otra persona contagiada por ella (caso secundario) presenta síntomas. Esta distribución, que también llamamos $\Phi(s)$ y que se denomina "serial interval" es más fácil de estimar experimentalmente y se adapta mejor al dato $i(t)$ observado del número de test positivos confirmados. Así y todo hay que indicar que no es perfecta pues habría que tener en cuenta el tiempo que pasa desde que una persona presenta síntomas hasta que se confirma su test positivo, lo cual es un dato muy variable que depende de la capacidad de hacer tests de los países y además no tiene en cuenta los casos asintomáticos. El "serial interval" $\Phi(s)$ tiene la propiedad de que puede ser mayor que cero para $s < 0$. Esto se produce porque, excepcionalmente, se puede dar que un caso secundario presente síntomas antes que el caso primario. Para aproximar la distribución $\Phi(s)$ se pueden usar diferentes tipos de distribuciones de probabilidad. Nosotros aquí utilizaremos la distribución log-normal que se utiliza con frecuencia en Epidemiología. La expresión general de la aproximación de $\Phi(s)$ por una distribución log-normal y su gráfica son las siguientes :

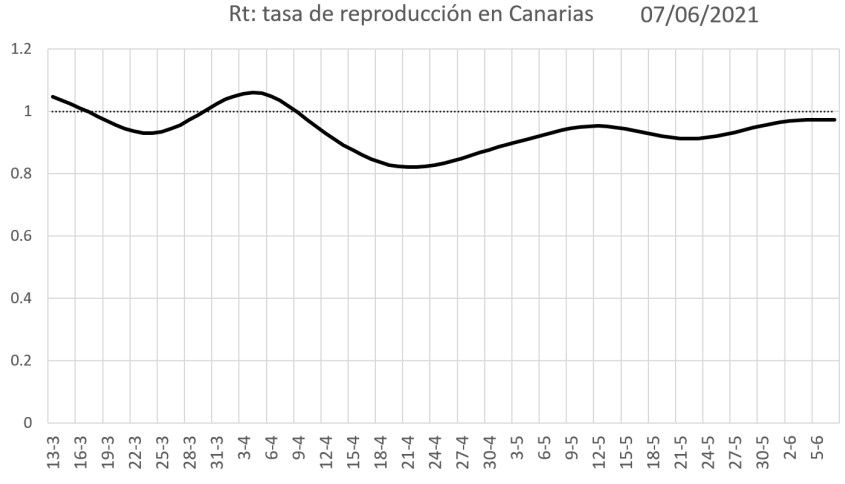
$$\Phi(x) = \begin{cases} \dfrac{e^{-\frac{(\ln(x-a)-b)^2}{2c}}}{(x-a)\sqrt{2\pi c}} & \text{si} \quad x > a \\ 0 & \text{si} \quad x \leq a \end{cases}$$



donde a, b y c son parámetros de la log-normal. En el caso de la COVID-19, como veremos en un próximo tema, el "serial interval" se aproxima razonablemente bien a través de la distribución log-normal usando como parámetros : $a = -4.979$, $b = 2.4918$ y $c = 0.1818$ que es la distribución que se muestra en la gráfica. Como puede apreciarse, esta distribución no es simétrica, sube con fuerza los primeros días para después bajar más suavemente y además $\Phi(s) > 0$ para $s > a = -4.979$, por tanto, el límite inferior de la integral de la "renewal equation" pasa a ser a (en lugar de 0) y la ecuación queda:

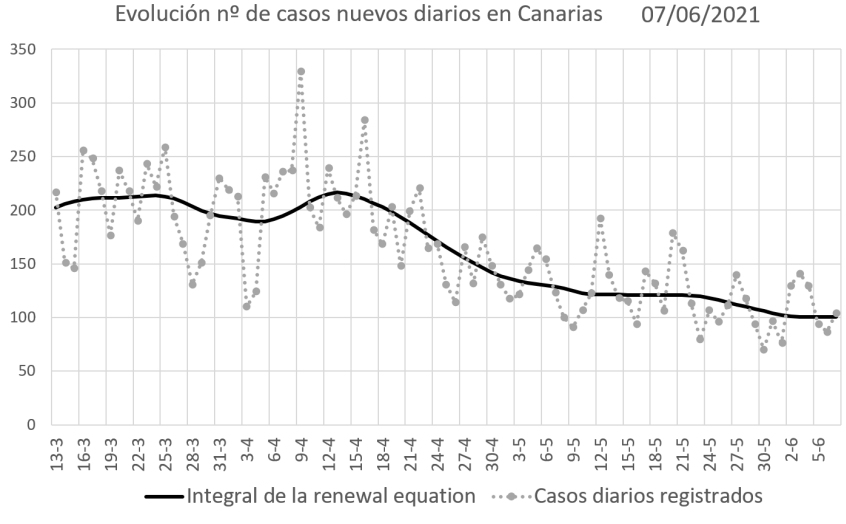
$$i(t) = \int_a^{Max(s)} i(t-s)R(t-s)\Phi(s)ds. \tag{5.61}$$

En la práctica, para aproximar numéricamente la evaluación de la "renewal equation", vamos a considerar que $\Phi(s) = 0$ si ha pasado ya un número grande de días desde la presentación de síntomas del caso primario. Por ejemplo, como se aprecia en la gráfica, a partir de 50 días (o incluso bastante antes), el valor de $\Phi(s)$ es tan pequeño que podemos considerarlo 0. Es decir, que podemos tomar $Max(s) = 50$ en la ecuacion. Nótese que si suponemos conocidos el número de casos diarios de nuevos test positivos confirmados, $i(t)$, la tasa de reproducción $R(t)$ y el "serial interval" $\Phi(s)$ podemos evaluar la integral de la derecha de la "renewal equation" usando las técnicas de integración numéricas vistas en este tema. $\Phi(s)$, definida a través de la distribución log-normal es una función continua definida para cualquier número real, sin embargo $i(t)$ y $R(t)$ solo se conocen en una colección de valores discretos (los días donde se calculan), por tanto, para evaluar estas funciones fuera de esos valores discretos hay que usar las técnicas de interpolación vistas en clase. En la siguiente gráfica se ilustra el valor de $R(t)$ calculado para Canarias usando el método propuesto en [ACMM].



En la práctica la "renewal equation" no se cumple de forma exacta debido a la imperfección del dato $i(t)$ observado que depende de muchos factores externos como

si el día es festivo o no, el número de test realizados y contabilizados, etc.. Con lo cual podemos interpretar la evaluación de la integral de la "renewal equation" como una estimación del valor esperado de $i(t)$. Usando el valor de $R(t)$, $i(t)$ y $\Phi(s)$ se puede calcular la integral de la "renewal equation" y compararlo con el valor de $i(t)$. Esto se muestra en la siguiente gráfica para Canarias usando el método de Simpson para aproximar la integral de la "renewal equation'.



como puede observarse en esta gráfica el evaluar la "renewal equation" nos da una versión suavizada del dato original $i(t)$ de nuevos casos diarios registrados.

Capítulo 6

ANÁLISIS NUMÉRICO MATRICIAL II Y OPTIMIZACIÓN

En este tema veremos algunos aspectos más avanzados del análisis matricial, incluyendo técnicas iterativas de resolución de sistemas de ecuaciones, cálculo de autovalores y resolución de sistemas no lineales. Además veremos una introducción a la optimización.

6.1. Normas de vectores y matrices.

Definición 4 *Una norma* $\| \ . \ \|$ *es una aplicación de un espacio vectorial* E *en* $R^+ \cup \{0\}$ *que verifica las siguientes propiedades:*

- $\| x \| = 0$ *si y sólo si* $x = 0$
- $\| \lambda x \| = | \lambda | \| x \|$ *para todo* $\lambda \in K$ *y* $x \in E$
- $\| x + y \| \leq \| x \| + \| y \|$ *para todo* $x, y \in E$.

Básicamente, una norma mide la magnitud o tamaño de un vector x. Por ejemplo, en el espacio vectorial de los números reales, la norma "natural" es el valor absoluto. Sin embargo, cuando trabajamos en varias dimensiones, esto es, $x = {}^t(x_1, x_2, ..., x_N)$, existen múltiples formas de definir una norma. La definición más utilizada es la denominada norma p, donde p es un número real positivo, que viene definida por

$$\| x \|_p = \left(\sum_{i=1}^{N} | x_i |^p \right)^{\frac{1}{p}}. \tag{6.1}$$

Un caso particularmente importante es $p = 2$, que corresponde a la norma Euclídea. Otro caso interesante es aquél que se produce cuando hacemos tender p hacia infinito, lo que da lugar a la denominada norma infinito definida por

$$\| x \|_\infty = max_i \mid x_i \mid . \tag{6.2}$$

Dada una matriz A de dimensión $N \times N$, se podría definir su norma considerando la matriz como un vector de dimensión $N \times N$. Sin embargo, resulta más útil definir la norma de una matriz subordinándola a la norma de un vector de la siguiente manera:

Definición 5 *Sea A una matriz y sea $\| . \|$ una norma vectorial. Se define la norma de A, subordinada a la norma vectorial $\| . \|$, como*

$$\| A \| = \sup_{x \neq 0} \frac{\| Ax \|}{\| x \|}. \tag{6.3}$$

La propiedad fundamental que verifica una norma matricial definida de esta forma es la siguiente:

Teorema 20 *Sea A una matriz y $\| . \|$ una norma vectorial. Entonces, para cualquier vector x se verifica que*

$$\| Ax \| \leq \| A \| \cdot \| x \| . \tag{6.4}$$

Demostración: Si $x = 0$, la desigualdad es trivial. Si $x \neq 0$, entonces, puesto que $\| x \| > 0$, la desigualdad anterior es equivalente a

$$\frac{\| Ax \|}{\| x \|} \leq \| A \| . \tag{6.5}$$

Ahora bien, esta desigualdad es cierta por la propia definición de $\| A \|$.

A continuación veremos la relación que existe entre la norma de una matriz y sus autovalores. Empezaremos recordando algunos conceptos relacionados con los autovalores.

Definición 6 *Un autovalor de A es un número real o complejo λ tal que existe un vector x, denominado autovector, verificando:*

$$Ax = \lambda x. \tag{6.6}$$

Definición 7 *Se denomina polinomio característico $P(\lambda)$ de la matriz A, al polinomio dado por el determinante*

$$P(\lambda) = \mid A - \lambda I \mid . \tag{6.7}$$

Definición 8 *Se define el radio espectral de una matriz A como*

$$\rho(A) = max_i\{\mid \lambda_i \mid : \ \lambda_i \text{ es autovalor de } A\}. \tag{6.8}$$

Teorema 21 *Sea A una matriz y* $\| \, . \, \|$ *una norma vectorial. Entonces*

$$\| A \| \geq \rho(A). \tag{6.9}$$

Demostración: Si λ es un autovalor de A, entonces existe un autovector x tal que $Ax = \lambda x$, por tanto

$$\frac{\| Ax \|}{\| x \|} = \frac{\| \lambda x \|}{\| x \|} = |\lambda| \leq \| A \|, \tag{6.10}$$

lo que demuestra el teorema.

Teorema 22 *Si los autovectores de una matriz A de dimensión* $N \times N$ *forman una base ortonormal de* R^N*, entonces*

$$\| A \|_2 = \rho(A).$$

Demostración: Recordamos, en primer lugar, que una base ortonormal de vectores es un conjunto de vectores tales que cualquier otro vector se puede expresar como combinación lineal de ellos y, además, su producto escalar verifica que

$$(x_i, x_j) = \sum_{k=1}^{N} (x_i)_k \, (x_j)_k = \begin{cases} 0 & si \quad i \neq j, \\ 1 & si \quad i = j, \end{cases} \tag{6.11}$$

donde $(x_i)_k$ indica la coordenada k-ésima del vector x_i. Vamos a demostrar la desigualdad

$$\| A \|_2 \leq \rho(A). \tag{6.12}$$

Dado que el teorema anterior determina la desigualdad en el otro sentido, tendríamos la igualdad, y por tanto el resultado del Teorema. Sea x un vector cualquiera. Puesto que A posee una base ortonormal de autovectores x_i, el vector x se podrá expresar como una combinación lineal de autovectores, de la forma:

$$x = \eta_1 x_1 + \eta_2 x_2 + .. + \eta_N x_N. \tag{6.13}$$

Al hacer Ax, y puesto que los x_i son autovectores, obtenemos que

$$Ax = \eta_1 \lambda_1 x_1 + \eta_2 \lambda_2 x_2 + .. + \eta_N \lambda_N x_N. \tag{6.14}$$

Como los autovectores son ortonormales, se cumple que

$$\begin{aligned} \| x \|_2 &= \sqrt{(\eta_1)^2 + .. + (\eta_N)^2}, \\ \| Ax \|_2 &= \sqrt{(\eta_1 \lambda_1)^2 + .. + (\eta_N \lambda_N)^2}. \end{aligned} \tag{6.15}$$

Y, por tanto,

$$\frac{\| Ax \|_2}{\| x \|_2} \leq \rho(A), \tag{6.16}$$

para cualquier vector x. En consecuencia, al tomar el supremo en x, la desigualdad se mantiene, lo que demuestra que

$$\| A \|_2 \leq \rho(A).$$

Teorema 23 *Si una matriz A de dimensión $N \times N$ es simétrica, entonces todos sus autovalores son reales y, además, R^N posee una base ortonormal formada por autovectores de A.*

Demostración: [La-Th].

Teorema 24 *Sea A una matriz cualquiera, entonces*

- $\| A \|_2 = \sqrt{\rho({}^tAA)}$,
- $\| A \|_1 = max_j \left(\sum_i | a_{ij} |\right)$,
- $\| A \|_\infty = max_i \left(\sum_j | a_{ij} |\right)$.

Demostración: [La-Th].

Teorema 25 *Sea A una matriz cualquiera, entonces*

$$Lim_{n\to\infty} \| A^n \| = 0 \Longleftrightarrow \rho(A) < 1. \tag{6.17}$$

Demostración:[La-Th].

6.2. Condicionamiento de una matriz.

El condicionamiento de una matriz es un número que nos indica la "bondad" o buen comportamiento numérico de la matriz cuando se trabaja con ella numéricamente para resolver sistemas. Para ilustrar de qué estamos hablando, veamos el siguiente ejemplo:

Ejemplo 12 *Consideremos el siguiente sistema de ecuaciones*

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ v \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}, \tag{6.18}$$

cuya solución es ${}^t(1,1,1,1)$. *Vamos a considerar ahora el mismo sistema, perturbando ligeramente el término independiente:*

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ v \end{pmatrix} = \begin{pmatrix} 32.1 \\ 22.9 \\ 33.1 \\ 30.9 \end{pmatrix}. \tag{6.19}$$

La solución de este sistema es ${}^t(9.2, -12.6, 4.5, -1.1)$. Como podemos observar, a pesar de que la perturbación del sistema es del orden de 0.1, la perturbación de la solución del sistema puede llegar a ser del orden de 13.6. El determinante de la matriz es 1, así que el valor del determinante no justifica este comportamiento.

Consideremos de forma genérica un sistema de ecuaciones de la forma

$$Au = b, \tag{6.20}$$

y, al mismo tiempo, el sistema de ecuaciones perturbado

$$A\left(u + \delta u\right) = b + \delta b. \tag{6.21}$$

Nosotros queremos controlar el error relativo en la solución del sistema a partir del error relativo en el término independiente b. Es decir, queremos encontrar una estimación del tipo

$$\frac{\| \delta u \|}{\| u \|} \leq \chi(A) \frac{\| \delta b \|}{\| b \|}, \tag{6.22}$$

donde $\chi(A)$ es un número que llamaremos condicionamiento de la matriz. Obviamente, cuanto más pequeño sea $\chi(A)$, mejor comportamiento numérico tendrá la matriz A.

Teorema 26 *Si definimos*

$$\chi(A) = \| A \| \cdot \| A^{-1} \|, \tag{6.23}$$

entonces

$$\frac{\| \delta u \|}{\| u \|} \leq \chi(A) \frac{\| \delta b \|}{\| b \|}. \tag{6.24}$$

Demostración: Como $A(u + \delta u) = b + \delta b$ y $Au = b$, se obtiene que $A\delta u = \delta b$, de donde $\delta u = A^{-1}\delta b$ y, por tanto,

$$\|\delta u\| \leq \left\|A^{-1}\right\| \|\delta b\|. \tag{6.25}$$

Por otro lado, también se cumple que

$$\|b\| = \|Au\| \leq \|A\| \|u\|, \tag{6.26}$$

de donde obtenemos que

$$\frac{1}{\|u\|} \leq \frac{\|A\|}{\|b\|}. \tag{6.27}$$

Así, multiplicando esta desigualdad con la anteriormente obtenida para $\|\delta u\|$, concluimos la demostración del teorema.

Nota: En el caso del ejemplo 12 los autovalores de la matriz son 0.01, 0.84, 3.86, y 30.29, por tanto el condicionamiento usando la norma 2 sería

$$\chi(A) = \frac{30.29}{0.01} = 3029, \tag{6.28}$$

lo cual indica un condicionamiento bastante malo.

6.3. Cálculo de autovalores y autovectores.

En esta sección veremos algunos métodos para el cálculo de autovalores y autovectores de matrices.

6.3.1. Método de Jacobi.

Este método se aplica a matrices reales y simétricas. Se basa en el hecho de que, dadas dos matrices A y R, se verifica que los autovalores de A son los mismos que los autovalores de $R^{-1}AR$. Este método intenta diagonalizar A realizando de forma iterativa transformaciones del tipo $R^{-1}AR$ para convertir en 0 los elementos de A fuera de la diagonal. Hay que tener en cuenta que si una matriz es diagonal (es decir fuera de la diagonal sus elementos son 0), sus autovalores son los elementos de la diagonal. Las matrices R que se utilizan son matrices de rotación, que tienen la forma siguiente:

$$R_{pq}(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & . & . & . & . & 0 \\ 0 & . & cos(\alpha) & . & sen(\alpha) & . & 0 \\ 0 & . & . & 1 & . & . & 0 \\ 0 & . & -sen(\alpha) & . & cos(\alpha) & . & 0 \\ 0 & . & . & . & . & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{6.29}$$

donde los cosenos y senos están situados en las columnas y filas p y q. Al ser una matriz de rotación, se verifica que $R_{pq}^{-1}(\alpha) = {}^tR_{pq}(\alpha)$ y por tanto si A es simétrica, la matriz ${}^tR_{pq}(\alpha)AR_{pq}(\alpha)$ también lo es. La matriz $R_{pq}(\alpha)$ se utiliza para hacer 0 los elementos $a_{p,q}$ y $a_{q,p}$. Al realizar la operación $A = {}^tR_{pq}(\alpha)AR_{pq}(\alpha)$, sólo se ven afectadas las filas y columnas de índices p y q. Concretamente la actualización de la matriz $A = {}^tR_{pq}(\alpha)AR_{pq}(\alpha)$ provoca los siguientes cambios en A :

Operaciones para la actualización de la matriz $A = {}^tR_{pq}(\alpha)AR_{pq}(\alpha)$

fila y columna p :

$$a_{k,p} = a_{p,k} = cos(\alpha)a_{p,k} - sen(\alpha)a_{q,k} \quad \forall\ k \neq p, q, \tag{6.30}$$

fila y columna q :

$$a_{k,q} = a_{q,k} = cos(\alpha)a_{q,k} + sen(\alpha)a_{p,k} \quad \forall\ k \neq p, q, \tag{6.31}$$

elementos diagonales

$$a_{p,p} = cos(\alpha)(cos(\alpha)a_{p,p} - sen(\alpha)a_{q,p}) - sen(\alpha)(cos(\alpha)a_{p,q} - sen(\alpha)a_{q,q}), \tag{6.32}$$

$$a_{q,q} = cos(\alpha)(cos(\alpha)a_{q,q} + sen(\alpha)a_{p,q}) + sen(\alpha)(cos(\alpha)a_{q,p} + sen(\alpha)a_{p,p}), \tag{6.33}$$

elementos $a_{p,q}$ y $a_{q,p}$ (que queremos convertir en 0)

$$a_{p,q} = a_{q,p} = a_{p,q}cos(2\alpha) - \frac{1}{2}\left(a_{q,q} - a_{p,p}\right) sen(2\alpha) = 0. \tag{6.34}$$

Al aplicar estas fórmulas para actualizar A hay que tener un poco de cuidado para no usar en los cálculos coeficientes ya actualizados de A.

Para convertir en cero los elementos $a_{p,q}$ y $a_{q,p}$ basta con despejar α de la expresión (6.34) obteniendo:

$$\tan(2\alpha) = \frac{2a_{pq}}{a_{qq} - a_{pp}} \quad \longrightarrow \quad \alpha = \frac{1}{2}\arctan\left(\frac{2a_{pq}}{a_{qq} - a_{pp}}\right), \tag{6.35}$$

en cada iteración del algoritmo se hace 0 el valor $a_{p,q}$ de mayor valor absoluto fuera de la diagonal.

Veamos ahora cómo podemos calcular los autovectores. Al utilizar el método de Jacobi, vamos transformando la matriz A multiplicándola por una secuencia de matrices de rotación $R_1, ..., R_M$, de tal forma que

$$R_M^{-1} \cdot .... \cdot R_1^{-1} A R_1 \cdot .... \cdot R_M = D, \tag{6.36}$$

donde D es una matriz diagonal que contiene los autovalores de A en la diagonal. Denotemos por R la matriz $R = R_1 \cdot .... \cdot R_M$. Despejando de la anterior igualdad obtenemos que

$$AR = RD, \tag{6.37}$$

lo cual nos indica que en la matriz R se encuentran almacenados los autovectores por columna y en la diagonal de D los correspondientes autovalores. Numéricamente, en cada iteración del algoritmo R se actualiza haciendo $R = R \cdot R_{p,q}(\alpha)$. Ahora bien,

dada la peculiar forma de las matrices de rotación, dicha actualización solo afecta a las columnas p y q de R de la siguiente manera:

$$\begin{aligned} R_{i,p} &= cos(\alpha)R_{i,p} - sen(\alpha)R_{i,q} \quad i = 0, .., N-1, \\ R_{i,q} &= sen(\alpha)R_{i,p} + cos(\alpha)R_{i,q} \quad i = 0, .., N-1. \end{aligned} \tag{6.38}$$

Además, R se inicializa al principio a la matriz identidad. Por tanto el algoritmo completo del método de Jacobi puede describirse de la siguiente manera:

Descripción del algoritmo de Jacobi para calcular autovalores y autovectores de matrices simétricas
Inicializamos la matriz de autovectores $R = Id$.
Para iter=0 **hasta** NmaxIter
Calcular p, q tal que $|a_{p,q}| = max_{j>i}\{|a_{i,j}|\}$
Si $|a_{p,q}| < TOL$ el algoritmo termina correctamente. Los autovalores están en la diagonal de A y los autovectores en la matriz R
Calcular $\alpha = 0.5 \cdot atan\left(\frac{2 \cdot a_{p,q}}{a_{q,q} - a_{p,p}}\right)$
Actualizar R haciendo $R = R \cdot R_{p,q}(\alpha)$
Actualizar la matriz A haciendo $A = {}^tR_{p,q}(\alpha) \cdot A \cdot R_{p,q}(\alpha)$
Fin Para
Se ha excedido el número de iteraciones.

Ejemplo 13 *Vamos a aplicar dos iteraciones del método de Jacobi para calcular los autovalores y autovectores de la matriz*

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}. \tag{6.39}$$

En la primera iteración convertimos en 0 el elemento $a_{01} = -1$ de la matriz, debemos elegir α tal que

$$\alpha = \frac{1}{2}\arctan\left(\frac{2a_{01}}{a_{11} - a_{00}}\right) = \frac{1}{2}\arctan\left(-\infty\right) = -\frac{\pi}{4}. \tag{6.40}$$

Por tanto, la matriz $R_{01}(\alpha)$ es

$$R_{01}(\alpha) = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{6.41}$$

al hacer la actualización $A = {}^tR_{01}(\alpha)AR_{01}(\alpha)$ obtenemos

$$A = {}^tR_{01}(\alpha)AR_{01}(\alpha) = \begin{pmatrix} 1 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 3 & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 2 \end{pmatrix}, \tag{6.42}$$

al actualizar R obtenemos

$$R = R \cdot R_{01}(\alpha) = R_{01}(\alpha). \tag{6.43}$$

En la segunda iteración hacemos 0 el elemento a_{02} de la matriz, debemos elegir α tal que

$$\alpha = \frac{1}{2}\arctan\left(\frac{2a_{02}}{a_{22} - a_{00}}\right) = \frac{1}{2}\arctan\left(-\frac{2}{\sqrt{2}}\right) = -0.4776583091. \tag{6.44}$$

Por tanto, la matriz $R_{02}(\alpha)$ es

$$R_{02}(\alpha) = \begin{pmatrix} cos(-0.4776583091) & 0 & sen(-0.4776583091) \\ 0 & 1 & 0 \\ -sen(-0.4776583091) & 0 & cos(-0.4776583091) \end{pmatrix}, \tag{6.45}$$

y al hacer la actualización $A = {}^tR_{02}(\alpha)AR_{02}(\alpha)$ obtenemos

$$A = {}^tR_{02}(\alpha)AR_{02}(\alpha) = \begin{pmatrix} 0.6339745963 & -0.3250575837 & 0 \\ -0.3250575837 & 3 & -0.6279630302 \\ 0 & -0.6279630302 & 2.366025404 \end{pmatrix}, \tag{6.46}$$

al actualizar R obtenemos

$$R = R \cdot R_{02}(\alpha) = \begin{pmatrix} 0.6781381579 & -2.351170765 & 0.4440369170 \\ 0.2184373144 & 1.891469922 & -0.4440369170 \\ 0 & -0.6279630302 & 2.366025404 \end{pmatrix}. \tag{6.47}$$

Por tanto, después de 2 iteraciones del algoritmo, los autovalores obtenidos serían la diagonal de A y sus autovectores los vectores columna de la matriz R. En este caso la aproximación es muy mala porque solo hemos dado 2 iteraciones del algoritmo y los elementos que quedan fuera de la diagonal de A están todavía muy lejos de 0.

Veamos ahora un resultado sobre la convergencia del método de Jacobi para el cálculo de autovalores.

Teorema 27 *Sea una matriz A simétrica. Sea $A^1 = A$, y sea A^k la matriz transformada de A^{k-1}, haciendo cero, (por el método de Jacobi) el elemento no diagonal mayor en módulo de la matriz A^{k-1}, entonces los elementos diagonales de la matriz A^k convergen ($k \to \infty$) hacia los autovalores de la matriz A. Además los elementos no diagonales de A convergen hacia* 0.

Demostración: [La-The].

6.3.2. Método de la potencia

El método de Jacobi tiene la limitación de que solo funciona para matrices simétricas. El método que vamos a estudiar ahora no requiere que la matriz sea simétrica y nos permite estimar el autovalor de mayor valor absoluto y su autovector asociado. Para introducir este método vamos a partir de un ejemplo, consideramos la matriz

$$A = \begin{pmatrix} -1 & 0 \\ -1 & -3 \end{pmatrix} \quad \longrightarrow \quad \lambda_{max} = -3, \qquad x_{max} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{6.48}$$

vamos a hacer iteraciones del esquema

$$u^n = \frac{Au^{n-1}}{\| u^{n-1} \|_\infty}, \tag{6.49}$$

partiendo de $u^0 = {}^t(1,1)$ obtenemos:

$$u^1 = \frac{Au^0}{\| u^0 \|_\infty} = \begin{pmatrix} -1 \\ -4 \end{pmatrix} \qquad \rightarrow \quad \| u^1 \|_\infty = 4, \tag{6.50}$$

$$u^2 = \frac{Au^1}{\| u^1 \|_\infty} = \begin{pmatrix} 1/4 \\ 13/4 \end{pmatrix} \qquad \rightarrow \quad \| u^2 \|_\infty = \frac{13}{4} = 3.25, \tag{6.51}$$

$$u^3 = \frac{Au^2}{\| u^2 \|_\infty} = \begin{pmatrix} -1/13 \\ -40/13 \end{pmatrix} \qquad \rightarrow \quad \| u^3 \|_\infty = \frac{40}{13} = 3.07, \tag{6.52}$$

$$\| u^1 \|_\infty = 4 \quad \| u^2 \|_\infty = 3.25 \quad \| u^3 \|_\infty = 3.07 \quad \rightarrow \quad 3, \tag{6.53}$$

$$\frac{u^1}{\| u^1 \|_\infty} = \begin{pmatrix} -1/4 \\ -1 \end{pmatrix}, \ \frac{u^2}{\| u^2 \|_\infty} = \begin{pmatrix} 1/13 \\ 1 \end{pmatrix}, \ \frac{u^3}{\| u^3 \|_\infty} = \begin{pmatrix} -1/40 \\ -1 \end{pmatrix}, \ \rightarrow \ \pm \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{6.54}$$

Observamos que $\| u^n \|_\infty$ converge hacia 3 que es el valor absoluto de λ_{max} y $\frac{u^n}{\|u^n\|_\infty}$ va alternando su signo y va convergiendo hacia $\pm x_{max}$. Como veremos formalmente a continuación, el método de la potencia se basa en que al hacer iteraciones del esquema (6.49) el vector resultante va hacia x_{max} alternando el signo si λ_{max} es negativo (no alterna el signo si $\lambda_{max} > 0$) y la norma del vector resultante va hacia $|\lambda_{max}|$. Además se puede usar cualquier norma vectorial para aplicar el método.

Teorema 28 *Sea una matriz A que posee una base de autovectores tal que en módulo su autovalor máximo λ_{max} es único. Sea un vector u^0 no ortogonal al subespacio engendrado por los autovectores del autovalor λ_{max}, entonces, si definimos la secuencia*

$$u^n = A\frac{u^{n-1}}{\| u^{n-1} \|}, \tag{6.55}$$

se verifica que

$$Lim_{n\to\infty} sign\left(\left(u^n, u^{n-1}\right)\right) \parallel u^n \parallel = \lambda_{max}, \tag{6.56}$$

$$Lim_{n\to\infty} \left(sign\left(\left(u^n, u^{n-1}\right)\right)\right)^n \frac{u^n}{\parallel u^n \parallel} \text{ es un autovector de } \lambda_{max}, \tag{6.57}$$

Además, dicho autovector tiene norma 1. $sign\left(\left(u^n, u^{n-1}\right)\right)$ *es el signo del producto escalar de* u^n *y* u^{n-1}, *es decir* $sign\left(\left(u^n, u^{n-1}\right)\right) = 1$ *si* $\left(u^n, u^{n-1}\right) \geq 0$ *y* $sign\left(\left(u^n, u^{n-1}\right)\right) = -1$ *si* $\left(u^n, u^{n-1}\right) < 0$.

Demostración. En primer lugar, vamos a demostrar por inducción la siguiente igualdad:

$$u^{n+1} = \frac{A^n u^1}{\|A^{n-1}u^1\|}. \tag{6.58}$$

Para $n = 1$ la igualdad se cumple por la definición de u^1. Supongamos que se cumple para $n-1$, y demostrémoslo para n:

$$u^{n+1} = A\frac{u^n}{\parallel u^n \parallel} = A\frac{\frac{A^{n-1}u^1}{\|A^{n-2}u^1\|}}{\frac{\|A^{n-1}u^1\|}{\|A^{n-2}u^1\|}} = \frac{A^n u^1}{\|A^{n-1}u^1\|}. \tag{6.59}$$

Con lo que queda demostrado este primer resultado. Por otro lado, como A posee una base de autovectores, que denotaremos por x_i, y u^1 no es ortogonal al espacio generado por los autovectores asociados a λ_{max}, entonces u^1 se puede escribir como

$$u^1 = \mu_1 x_1 + ... + \mu_N x_N, \tag{6.60}$$

donde supondremos que x_1 es un autovector asociado a λ_{max} y que $\mu_1 \neq 0$. Por la igualdad anteriormente demostrada obtenemos que

$$u^n = \frac{A^{n-1}u^1}{\|A^{n-2}u^1\|} = \frac{\mu_1\lambda_{max}^{n-1}x_1 + ... + \mu_N\lambda_N^{n-1}x_N}{\left\|\mu_1\lambda_{max}^{n-2}x_1 + ... + \mu_N\lambda_N^{n-2}x_N\right\|} = \tag{6.61}$$

$$= |\lambda_{max}| \frac{\mu_1\left(\frac{\lambda_{max}}{|\lambda_{max}|}\right)^{n-1}x_1 + ... + \mu_N\left(\frac{\lambda_N}{|\lambda_{max}|}\right)^{n-1}x_N}{\left\|\mu_1\left(\frac{\lambda_{max}}{|\lambda_{max}|}\right)^{n-2}x_1 + ... + \mu_N\left(\frac{\lambda_N}{|\lambda_{max}|}\right)^{n-2}x_N\right\|}. \tag{6.62}$$

Cuando hacemos tender n hacia infinito, todos los cocientes de la forma

$$\left(\frac{\lambda_i}{|\lambda_{max}|}\right)^n, \tag{6.63}$$

tienden hacia 0, salvo si $\lambda_i = \lambda_{max}$. En este caso, dicho cociente es 1^n, si λ_{max} es positivo, o $(-1)^n$, si λ_{max} es negativo. Por tanto, para n suficientemente grande el signo de λ_{max} coincide con el signo del producto escalar (u^n, u^{n-1}). Además

$$Lim_{n\to\infty}\|u^n\| = |\lambda_{max}| \frac{\left\|\mu_1\left(\frac{\lambda_{max}}{|\lambda_{max}|}\right)^{n-1}x_1 + ... + \mu_N\left(\frac{\lambda_N}{|\lambda_{max}|}\right)^{n-1}x_N\right\|}{\left\|\mu_1\left(\frac{\lambda_{max}}{|\lambda_{max}|}\right)^{n-2}x_1 + ... + \mu_N\left(\frac{\lambda_N}{|\lambda_{max}|}\right)^{n-2}x_N\right\|} = |\lambda_{max}|\,, \tag{6.64}$$

y por tanto

$$Lim_{n\to\infty} sign\left((u^n, u^{n-1})\right) \parallel u^n \parallel = \lambda_{max}. \tag{6.65}$$

Por otro lado, el término $\left(sign\left((u^n, u^{n-1})\right)\right)^n$ para n suficientemente grande es 1^n si λ_{max} es positivo o $(-1)^n$ si λ_{max} es negativo. Sean $x_1, ..., x_M$ los autovectores asociados a λ_{max}, obtenemos que

$$Lim_{n\to\infty}\left(sign\left((u^n, u^{n-1})\right)\right)^n \frac{u^n}{\parallel u^n \parallel} = \frac{\mu_1 x_1 + ... + \mu_M x_M}{\|\mu_1 x_1 + ... + \mu_M x_M\|}, \tag{6.66}$$

que es un autovector de λ_{max} de norma 1.

Descripción algoritmo de la potencia

- Se parte de una aproximación inicial del autovector máximo u.
- Se inicia un procedimiento iterativo que actualiza u y el autovalor λ.
  - Se calcula Au y su norma $\lambda = \parallel Au \parallel$.
  - Si hay cambio de signo entre Au y u se cambia el signo de λ.
  - Se actualiza $u = u/\lambda$.
  - El criterio de parada es que la norma de u menos su versión anterior sea menor que una tolerancia TOL, o que se supere el número de iteraciones.

6.3.3. Método de la potencia inversa.

El método anterior también se puede utilizar para el cálculo del autovalor de A de módulo menor λ_{min} teniendo en cuenta que

$$max\{\lambda_i' \ : autovalores\ de\ A^{-1}\} = \frac{1}{\lambda_{min}}. \tag{6.67}$$

Si aplicamos el método anterior a A^{-1}, obtenemos que la secuencia

$$u^n = A^{-1}\frac{u^{n-1}}{\parallel u^{n-1} \parallel}, \tag{6.68}$$

verifica que

$$Lim_{n\to\infty} sign\left(\left(u^n, u^{n-1}\right)\right) \parallel u^n \parallel = \frac{1}{\lambda_{min}}, \tag{6.69}$$

$$Lim_{n\to\infty} sign\left(\left(u^n, u^{n-1}\right)\right) \frac{u^n}{\parallel u^n \parallel} \text{ es un autovector de } \lambda_{min}. \tag{6.70}$$

En los casos prácticos, se puede evitar calcular A^{-1} y obtener u^n resolviendo el sistema

$$Au^n = \frac{u^{n-1}}{\parallel u^{n-1} \parallel}. \tag{6.71}$$

Los métodos de la potencia directa e inversa nos permiten calcular el autovalor más grande y más pequeño de una matriz. Ahora bien, para calcular otros autovalores de la matriz A necesitamos información adicional como puede ser tener una aproximación del autovalor buscado. Si μ es una aproximación de un autovalor λ, de tal forma que μ se encuentre más cerca de λ que de cualquier otro autovalor de A, podemos construir la matriz $A' = A - \mu I$ y calcular su autovalor mínimo λ'_{min} por el método de la potencia inversa obteniendo:

$$A'\bar{x} = (A - \mu I)\bar{x} = \lambda'_{min}\bar{x}, \tag{6.72}$$

de donde despejando

$$A\bar{x} = (\lambda'_{min} + \mu)\bar{x}, \tag{6.73}$$

y por tanto $\lambda'_{min} + \mu$ es el autovalor de A más cercano a μ.

El razonamiento anterior nos lleva al siguiente algoritmo general para calcular todos los autovectores y autovalores de una matriz cualquiera:

Algoritmo para calcular todos los autovalores de una matriz usando el método de la potencia inversa

1. Paso 1: Se calcula el polinomio característico $|A - \lambda I| = 0$.
2. Paso 2: Se calculan las raíces $\{\lambda_i\}$ del polinomio característico.
3. Paso 3 : Utilizando el método de la potencia inversa se calculan los autovectores más pequeños de las matrices $A - \lambda_i I$.

En el paso 1, para calcular el polinomio característico se puede proceder de la siguiente manera: Dada una matriz A de dimensión N se calcula $\|A\|_1$, lo cual es muy rápido pues sólo hay que sumar elementos de las columnas de la matriz y calcular su máximo, como todos los autovalores de A cumplen que $|\lambda_i| \leq \|A\|_1$, ello significa que todos los autovalores están en el intervalo $[-\|A\|_1, \|A\|_1]$. Elegimos a continuación $N+1$ valores x_i equidistantes en $[-\|A\|_1, \|A\|_1]$. Para cada uno de esos valores tenemos que

$$|A - x_i I| = a_N x_i^N + a_{N-1} x_i^{N-1} + ... + a_0, \tag{6.74}$$

donde $\{a_N, a_{N-1}, ..., a_0\}$ representan los coeficientes del polinomio característico buscado. Nótese que para x_i la relación de arriba es una ecuación lineal en los coeficientes del polinomio y por tanto dichos coeficientes se pueden encontrar resolviendo el sistema:

$$\begin{pmatrix} x_0^N & x_0^{N-1} & . & 1 \\ x_1^N & x_1^{N-1} & . & 1 \\ . & . & . & . \\ x_N^N & x_N^{N-1} & . & 1 \end{pmatrix} \begin{pmatrix} a_N \\ a_{N-1} \\ . \\ a_0 \end{pmatrix} = \begin{pmatrix} |A - x_0 I| \\ |A - x_1 I| \\ . \\ |A - x_N I| \end{pmatrix}. \tag{6.75}$$

6.4. Métodos iterativos de resolución de sistemas lineales.

Estas técnicas consisten en transformar un sistema de la forma

$$Au = b, \tag{6.76}$$

en una ecuación de punto fijo de la forma

$$u = Mu + c, \tag{6.77}$$

de tal manera que, al hacer iteraciones de la forma

$$u^n = Mu^{n-1} + c, \tag{6.78}$$

se obtenga que u^n converge hacia u, la solución del sistema original.

Ejemplo 14 *Consideremos el sistema de ecuaciones*

$$\begin{aligned} 2x - y &= 1, \\ -x + 2y - z &= 0, \\ -y + 2z &= 1. \end{aligned} \tag{6.79}$$

Despejando la diagonal del sistema de ecuaciones obtenemos que buscar la solución de este sistema es equivalente a buscar un vector $u = {}^t(x, y, z)$ *que verifique que*

$$\begin{aligned} x &= \frac{1+y}{2}, \\ y &= \frac{x+z}{2}, \\ z &= \frac{1+y}{2}. \end{aligned} \tag{6.80}$$

Hacer iteraciones de esta ecuación de punto fijo consiste en partir de una aproximación inicial $u^0 = {}^t(x_0, y_0, z_0)$ y hacer iteraciones de la forma

$$\begin{aligned} x_n &= \frac{1 + y_{n-1}}{2}, \\ y_n &= \frac{x_{n-1} + z_{n-1}}{2}, \\ z_n &= \frac{1 + y_{n-1}}{2}. \end{aligned} \tag{6.81}$$

En este caso, la solución exacta del sistema es $u = {}^t(1, 1, 1)$. Si hacemos iteraciones del esquema anterior a partir de la aproximación inicial $u^0 = {}^t(0, 0, 0)$, obtenemos que

$$\begin{aligned} x_1 &= \frac{1+0}{2} = \frac{1}{2}, \\ y_1 &= \frac{0+0}{2} = 0, \\ z_1 &= \frac{1+0}{2} = \frac{1}{2}. \end{aligned} \tag{6.82}$$

De la misma forma, obtenemos que

$$u^2 = \begin{pmatrix} 0.5 \\ 0.25 \\ 0.5 \end{pmatrix}, \ldots. \; u^7 = \begin{pmatrix} 0.84 \\ 0.73 \\ 0.84 \end{pmatrix}, \ldots. \; u^{16} = \begin{pmatrix} 0.98 \\ 0.96 \\ 0.98 \end{pmatrix}. \tag{6.83}$$

Como puede observarse, las sucesivas iteraciones se van aproximando a la solución $u = {}^t(1, 1, 1)$. En este caso, la matriz M y el vector c que determinan el esquema iterativo vienen dados por

$$M_J = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 \end{pmatrix}, \quad c_J = \begin{pmatrix} \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{pmatrix}. \tag{6.84}$$

Teorema 29 *Si el esquema iterativo*

$$u^n = Mu^{n-1} + c, \tag{6.85}$$

converge hacia un vector u, entonces u verifica que

$$u = Mu + c. \tag{6.86}$$

Demostración: es inmediata teniendo en cuenta que si u^n converge hacia u entonces Mu^{n-1} converge hacia Mu.

Existen diferentes métodos para convertir un sistema de la forma $Au = b$ en una ecuación de punto fijo $u = Mu + c$. Todas se basan en descomponer A de la forma

$A = L + D + U$, donde D es la matriz diagonal que corresponde a la parte diagonal de A, L es la matriz triangular inferior que corresponde a la parte de A situada por debajo de la diagonal, y U es la matriz triangular superior que corresponde a la parte de A situada por encima de la diagonal.

6.4.1. Método de Jacobi

Este método consiste en tomar

$$M_J = D^{-1}(-L - U), \tag{6.87}$$

$$c_J = D^{-1}b, \tag{6.88}$$

que es la descomposición que se ha utilizado en el ejemplo anterior. El paso de una iteración a otra del método de Jacobi puede expresarse de la siguiente forma:

$$\begin{aligned} u_1^n &= \frac{-a_{12}u_2^{n-1} - ... - a_{1N}u_N^{n-1} + b_1}{a_{11}}, \\ u_2^n &= \frac{-a_{21}u_1^{n-1} - a_{23}u_3^{n-1} ... - a_{2N}u_N^{n-1} + b_2}{a_{22}}, \\ &. \\ u_N^n &= \frac{-a_{N1}u_1^{n-1} - a_{N2}u_2^{n-1} ... - a_{NN-1}u_{N-1}^{n-1} + b_N}{a_{NN}}. \end{aligned} \tag{6.89}$$

6.4.2. Método de Gauss-Seidel

Este método consiste en tomar

$$M_{GS} = (D + L)^{-1}(-U), \tag{6.90}$$

$$c_{GS} = (D + L)^{-1} b. \tag{6.91}$$

A efectos prácticos, la aplicación de este método no requiere el cálculo de la matriz inversa $(D + L)^{-1}$ puesto que el paso de una iteración a otra puede hacerse de la siguiente forma:

$$\begin{aligned} u_1^n &= \frac{-a_{12}u_2^{n-1} - ... - a_{1N}u_N^{n-1} + b_1}{a_{11}}, \\ u_2^n &= \frac{-a_{21}u_1^{n} - a_{23}u_3^{n-1} ... - a_{2N}u_N^{n-1} + b_2}{a_{22}}, \\ &. \\ u_N^n &= \frac{-a_{N1}u_1^{n} - a_{N2}u_2^{n} ... - a_{NN-1}u_{N-1}^{n} + b_N}{a_{NN}}. \end{aligned} \tag{6.92}$$

Efectivamente, si hacemos un barrido para el cálculo de la solución de arriba hacia abajo, y vamos actualizando las componentes del vector aproximación según

las vamos calculando, obtenemos el método de Gauss-Seidel. Por tanto, básicamente, podemos decir que la diferencia entre el método de Gauss-Seidel y el método de Jacobi es que en el método de Gauss-Seidel se actualiza el vector aproximación después del cálculo de cada componente, y en el caso de Jacobi se actualiza sólo al final, después de haber calculado todas las componentes por separado.

Ejemplo 15 *Vamos a aplicar el método de Gauss-Seidel al sistema del ejemplo anterior, es decir*

$$\begin{aligned} 2x - y &= 1, \\ -x + 2y - z &= 0, \\ -y + 2z &= 1. \end{aligned} \tag{6.93}$$

Las iteraciones del método de Gauss-Seidel aplicado a este sistema consisten en

$$\begin{aligned} x_n &= \frac{1 + y_{n-1}}{2}, \\ y_n &= \frac{x_n + z_{n-1}}{2}, \\ z_n &= \frac{1 + y_n}{2}. \end{aligned} \tag{6.94}$$

Si hacemos iteraciones del esquema anterior a partir de la aproximación inicial $u^0 = {}^t(0,0,0)$, obtenemos que

$$\begin{aligned} x_1 &= \frac{1+0}{2} = \frac{1}{2}, \\ y_1 &= \frac{\frac{1}{2}+0}{2} = \frac{1}{4}, \\ z_1 &= \frac{1+\frac{1}{4}}{2} = \frac{5}{8}. \end{aligned} \tag{6.95}$$

De la misma forma, las siguientes iteraciones del esquema son:

$$u^2 = \begin{pmatrix} 0.5 \\ 0.25 \\ 0.625 \end{pmatrix}, ..., \; u^8 = \begin{pmatrix} 0.97656 \\ 0.97656 \\ 0.98828 \end{pmatrix}. \tag{6.96}$$

En el método de Gauss-Seidel, al actualizar las incógnitas, las últimas se ven favorecidas debido a que utilizan información ya actualizada de las incógnitas anteriores. Para compensar este efecto se puede alternar la dirección de los barridos haciendo en una iteración el barrido de actualizaciones de arriba a abajo y en la siguiente iteración de abajo a arriba.

6.4.3. Método de relajación

El objetivo de este método es intentar mejorar la velocidad del método de Gauss-Seidel introduciendo un parámetro de relajación w. Se toman, en este caso,

$$\begin{aligned} M_w &= (D + wL)^{-1} ((1 - w)D - wU), \\ c_w &= w (D + wL)^{-1} b. \end{aligned} \tag{6.97}$$

Estas nuevas matrices permiten realizar un promediado entre el resultado obtenido por Gauss-Seidel y el estado de la solución en la etapa anterior de la forma siguiente:

$$\begin{aligned} u_1^n &= w\frac{-a_{12}u_2^{n-1} - ... - a_{1N}u_N^{n-1} + b_1}{a_{11}} + (1 - w)u_1^{n-1}, \\ u_2^n &= w\frac{-a_{21}u_1^n ... - a_{2N}u_N^{n-1} + b_2}{a_{22}} + (1 - w)u_2^{n-1}, \\ &\cdot \\ u_N^n &= w\frac{-a_{N1}u_1^n ... - a_{NN-1}u_{N-1}^n + b_N}{a_{NN}} + (1 - w)u_N^{n-1}. \end{aligned} \tag{6.98}$$

La elección del parámetro w es, en general, un problema difícil. Sin embargo, en el caso de matrices tridiagonales, es decir, matrices con todos los elementos nulos salvo la diagonal principal y sus codiagonales, el siguiente resultado muestra la forma de calcular el valor óptimo de w.

Teorema 30 *Si A es una matriz tridiagonal y $\rho(M_J) < 1$, entonces el valor de w que optimiza la velocidad de convergencia del método es:*

$$w_{opt} = \frac{2}{1 + \sqrt{1 - \rho(M_J)^2}}. \tag{6.99}$$

Como puede observarse de la expresión anterior, el valor de w_{opt} se encuentra siempre entre 1 y 2.
Demostración [La-Th].

Ejemplo 16 *Vamos a aplicar el método de relajación al sistema del ejemplo anterior, es decir*

$$\begin{aligned} 2x - y &= 1, \\ -x + 2y - z &= 0, \\ -y + 2z &= 1. \end{aligned} \tag{6.100}$$

En este caso, $\rho(M_J) = \frac{1}{\sqrt{2}}$ y $w_{opt} = 1.17$. Las iteraciones del método de relajación aplicado a este sistema consisten en hacer

$$\begin{aligned} x_n &= w\frac{1+y_{n-1}}{2} + (1-w)x_{n-1}, \\ y_n &= w\frac{x_n+z_{n-1}}{2} + (1-w)y_{n-1}, \\ z_n &= w\frac{1+y_n}{2} + (1-w)z_{n-1}. \end{aligned} \tag{6.101}$$

Si hacemos iteraciones del esquema anterior a partir de la aproximación inicial $u^0 = {}^t(0,0,0)$ y tomando $w = w_{opt} = 1.17$, obtenemos que

$$u^1 = \begin{pmatrix} 0.585 \\ 0.342 \\ 0.785 \end{pmatrix},\ u^2 = \begin{pmatrix} 0.686 \\ 0.802 \\ 0.921 \end{pmatrix}, ...,\ u^7 = \begin{pmatrix} 0.999 \\ 0.999 \\ 0.999 \end{pmatrix}. \tag{6.102}$$

6.4.4. Convergencia de los métodos iterativos.

Vamos a denotar por $e^n = u^n - u$ el error relativo entre la solución del sistema u y la aproximación en la etapa n, u^n.

Teorema 31 *Se considera el esquema iterativo $u^n = Mu^{n-1} + c$. Entonces*

$$e^n = M^{n-1}e^1.$$

Demostración: La solución del sistema satisface que $u = Mu + c$. Restando esta igualdad de la igualdad $u^n = Mu^{n-1} + c$, obtenemos que

$$u^n - u = M(u^{n-1} - u) = M^{n-1}(u^1 - u). \tag{6.103}$$

Teorema 32 *El método iterativo $u^n = Mu^{n-1} + c$ converge para cualquier aproximación inicial si y sólo si $\rho(M) < 1$.*

Demostración: El resultado es inmediato a partir del hecho de que una matriz M^n converge hacia 0 cuando $n \to \infty$ si y sólo si $\rho(M) < 1$.

Teorema 33 *Si en el método de relajación $w \notin (0,2)$, entonces $\rho(M_w) \geq 1$.*

Demostración: En primer lugar, observamos que las matrices $D + Lw$ y $(1-w)D - wU$ son matrices triangulares y, por tanto, su determinante es el producto de los elementos diagonales. Además, teniendo en cuenta que el determinante del producto de dos matrices es el producto de sus determinantes y que el determinante de la matriz inversa es el inverso del determinante, obtenemos que

$$|M_w| = \frac{|(1-w)D - wU|}{|(D+wL)|} = \frac{(1-w)^N \Pi_i a_{ii}}{\Pi_i a_{ii}} = (1-w)^N. \tag{6.104}$$

Por lo tanto, como el determinante de una matriz es el producto de sus autovalores, obtenemos que, si $w \notin (0,2)$, entonces $|1-w| \geq 1$ y, en consecuencia, M_w posee al menos un autovalor de módulo mayor o igual que uno.

Teorema 34 *Si una matriz A verifica que*

$$| a_{ii} | > \sum_{j \neq i} | a_{ij} | \quad \forall i, \qquad o \qquad | a_{jj} | > \sum_{i \neq j} | a_{ij} | \quad \forall j. \tag{6.105}$$

entonces el método de Jacobi asociado al sistema $Au = b$ converge para cualquier aproximación inicial.

Demostración: En primer lugar, observamos que la matriz M_J puede expresarse como:

$$\begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & . & -\frac{a_{1N}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & . & -\frac{a_{2N}}{a_{22}} \\ . & . & . & . & . \\ -\frac{a_{N-1,1}}{a_{N-1,N-1}} & -\frac{a_{N-1,2}}{a_{N-1,N-1}} & . & 0 & -\frac{a_{N-1,N}}{a_{N-1,N-1}} \\ -\frac{a_{N,1}}{a_{N,N}} & -\frac{a_{N,2}}{a_{N,N}} & . & -\frac{a_{N,N-1}}{a_{N,N}} & 0 \end{pmatrix}. \tag{6.106}$$

Teniendo en cuenta que las normas 1 e infinito de una matriz son el máximo de las sumas por filas o columnas en valor absoluto, se tiene, por las condiciones del teorema, que $\|M_J\| < 1$ para la norma 1 o infinito. Por tanto, el teorema se concluye teniendo en cuenta que cualquier norma de una matriz es siempre mayor o igual que su radio espectral. Este resultado se puede generalizar un poco al caso de matrices irreducibles de la siguiente forma:

Definición 9 *Una matriz A es irreducible si un sistema de la forma $Au = b$ no puede descomponerse en dos subsistemas independientes de dimensión menor*

Dicho de otra forma, una matriz es irreducible si el cambio de cualquier valor del vector b del sistema $Au = b$ puede afectar a todos los elementos del vector u.

Teorema 35 *Si A es una matriz irreducible y se verifica que*

$$| a_{ii} | \geq \sum_{j \neq i} | a_{ij} | \quad \forall i, \qquad o \qquad | a_{jj} | \geq \sum_{i \neq j} | a_{ij} | \quad \forall j. \tag{6.107}$$

con la desigualdad estricta en al menos una fila o columna, entonces los métodos iterativos convergen.

Demostración. [La-The].

Ejemplo 17 *La matriz del sistema ejemplo tratado anteriormente, esto es*

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}, \tag{6.108}$$

satisface las hipótesis del teorema anterior.

6.4.5. Matrices escasas

Los métodos iterativos están especialmente indicados en el caso de matrices escasas que son aquellas que contienen muchos coeficientes iguales a cero. Estas matrices aparecen con frecuencia en las aplicaciones prácticas como cálculos de estructuras de edificios, modelización del clima, etc.. Para almacenar estas matrices solo se guardan los elementos distintos de cero. Los elementos se guardan por filas. Cada fila tiene un tamaño variable en función de los elementos no nulos en cada fila. Además, para acelerar la ejecución de los métodos iterativos guardaremos el elemento de la diagonal como primer elemento de cada fila. Para completar la información, además de los elementos no nulos tenemos que almacenar el índice de la columna donde se encuentra el elemento no nulo. Para ello utilizaremos una matriz de índices que llamaremos J. A continuación veremos un ejemplo comparativo de como se almacena una matriz en forma plena (es decir almacenando todos los elementos) y en forma escasa:

Almacenamiento usual de una matriz en forma plena

$A = \begin{pmatrix} 1 & 0 & 5 \\ 3 & 2 & 0 \\ 0 & 0 & 6 \end{pmatrix}$	Recorrido de la matriz A : $A[i][j]$ $0 < i < N$ y $0 < j < M$

Almacenamiento de la misma matriz en forma escasa

$A = \begin{pmatrix} 1 & 5 \\ 2 & 3 \\ 6 & \end{pmatrix}$ $J = \begin{pmatrix} 0 & 2 \\ 1 & 0 \\ 2 & \end{pmatrix}$	Recorrido de la matriz A : $A[i][j] \rightarrow A[i][J[i][j]]$ $0 < i < N$ $0 < j < J[i].dim()$

donde $J[i].\dim()$ representa el número de elementos no nulos de la fila i. A continuación presentamos un algoritmo para multiplicar una matriz almacenada en forma escasa A de $A.\dim()$ filas por un vector b y el resultado se almacena en un vector u.

Algoritmo multiplicación matriz escasa vector

```
para i=0 hasta A.dim()-1 hacer
   u[i]=0;
   para j=0 hasta J[i].dim()-1 hacer
      u[i]= u[i]+A[i][j]* b[J[i][j]];
   fin para
fin para
```

Método de Relajación con matrices escasas

Para el método de relajación, a continuación comparamos como se actualiza en cada iteración el vector solución usando matrices normales o escasas (suponemos que en la matriz escasa el primer valor de cada fila corresponde a la diagonal)

Actualización en cada iteración de cada elemento del vector solución u usando el método con matrices almacenadas de la forma usual

$$u[i] = w\frac{-\sum_{j\neq i} A[i][j] * u[j] + b[i]}{A[i][i]} + (1-w) * u[i]. \tag{6.109}$$

Actualización en cada iteración de cada elemento del vector solución u usando el método con matrices almacenadas en forma escasa

$$u[i] = w\frac{-\sum_{j=1}^{J[i].\dim()-1} A[i][j] * u[J[i][j]] + b[i]}{A[i][0]} + (1-w) * u[i], \tag{6.110}$$

como puede observarse, en el caso de matrices escasas la actualización de la solución por los métodos iterativos es más sencilla y rápida que hacerlo para una matriz plena porque solo se hacen operaciones con los elementos no nulos de la matriz. Sin embargo, este tipo de almacenamiento no es adecuado para los métodos directos como Gauss, etc.. porque en estos métodos los elementos no nulos de las matrices van cambiando durante el proceso y eso provoca que haya que estar constantemente actualizando la memoria y contenido de la matriz escasa lo cual es ineficiente computacionalmente.

6.5. Método de Newton-Raphson para sistemas no lineales

En las aplicaciones reales, muchas veces nos encontramos con sistemas no lineales de ecuaciones. Por ejemplo, calcular las raíces, reales o complejas, de un polinomio de grado 2 dado por $P_2(z) = az^2 + bz + c$, donde $z = x + yi$, es equivalente a resolver el sistema

$$\begin{aligned} ax^2 + bx - ay^2 + c &= 0, \\ 2ayx + by &= 0, \end{aligned} \tag{6.111}$$

que es un sistema no lineal de ecuaciones. En general, un sistema no lineal de ecuaciones de dimensión N se escribe como N ecuaciones del tipo

$$\begin{aligned} f_1(u_0, ..., u_{N-1}) &= 0, \\ f_2(u_0, ..., u_{N-1}) &= 0, \\ &\cdot \\ f_N(u_0, ..., u_{N-1}) &= 0, \end{aligned} \tag{6.112}$$

donde $f(u) =^t (f_0(u), f_2(u), ..., f_{N-1}(u))$ es una función de $R^N \to R^N$, y $u =^t (u_0, ..., u_{N-1})$. El método de Newton-Raphson para sistemas no-lineales se basa en desarrollar por Taylor la función f y truncar el desarrollo para que quede un sistema lineal, es decir

$$f(u) = f(u^0) + \nabla f(u^0)\left(u - u^0\right) + \mathcal{O}\left(\| u - u^0 \|^2\right), \tag{6.113}$$

donde u^0 es una aproximación de la solución de $f(u) = 0$. Si truncamos el desarrollo e igualamos a 0 (para aproximar la raíz) obtenemos que la raíz del sistema lineal se obtiene resolviendo el sistema

$$\begin{aligned} \nabla f(u^0) z &= -f(u^0), \\ u^1 &= u^0 + z. \end{aligned} \tag{6.114}$$

Por tanto, haciendo iteraciones de este procedimiento obtenemos el esquema:

$$\begin{aligned} \nabla f(u^n) z &= -f(u^n), \\ u^{n+1} &= u^n + z. \end{aligned} \tag{6.115}$$

Ejemplo 18 *Consideremos el siguiente sistema no lineal de ecuaciones:*

$$\begin{aligned} x^2 - y^2 + 1 &= 0, \\ 2xy &= 0. \end{aligned} \tag{6.116}$$

La matriz gradiente de esta función viene dada por

$$\nabla f(u^n) = \begin{pmatrix} 2x & -2y \\ 2y & 2x \end{pmatrix}. \tag{6.117}$$

Tomemos como aproximación inicial $u^0 =^t (1, 1)$. El sistema que hay que resolver para pasar de una iteración a otra es

$$\begin{pmatrix} 2x_n & -2y_n \\ 2y_n & 2x_n \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = - \begin{pmatrix} x_n^2 - y_n^2 + 1 \\ 2y_n x_n \end{pmatrix}. \tag{6.118}$$

Si partimos de $u^0 =^t (1, 1)$, para obtener u^1 tenemos que resolver

$$\begin{pmatrix} 2 & -2 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \end{pmatrix}, \tag{6.119}$$

que tiene por solución $^t\left(-\frac{3}{4}, -\frac{1}{4}\right)$. Por tanto, u^1 viene dado por

$$u^1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} -\frac{3}{4} \\ -\frac{1}{4} \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ \frac{3}{4} \end{pmatrix}. \tag{6.120}$$

Para calcular u^2, tenemos que resolver el sistema

$$\begin{pmatrix} \frac{1}{2} & -\frac{3}{2} \\ \frac{3}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = - \begin{pmatrix} \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 + 1 \\ \frac{6}{16} \end{pmatrix}, \tag{6.121}$$

cuya solución es ${}^t\left(-\frac{13}{40}, -\frac{9}{40}\right)$. Por tanto, u^2 viene dado por

$$u^2 = \begin{pmatrix} \frac{1}{4} \\ \frac{3}{4} \end{pmatrix} + \begin{pmatrix} -\frac{13}{40} \\ \frac{9}{40} \end{pmatrix} = \begin{pmatrix} -\frac{3}{40} \\ \frac{39}{40} \end{pmatrix}, \tag{6.122}$$

que ya es una buena aproximación de la solución exacta dada por el vector ${}^t(0,1)$.

Para implementar el método de Newton-Raphson para sistemas lo primero que tenemos que hacer es aproximar numéricamente la matriz gradiente dada por

$$\nabla f(u) = \left(\frac{\partial f}{\partial u_0}(u), .., \frac{\partial f}{\partial u_k}(u), .., \frac{\partial f}{\partial u_{N-1}}(u)\right), \tag{6.123}$$

es decir cada columna de la matriz $\nabla f(u)$ es una derivada parcial de $f(u)$. Estas derivadas parciales las podemos aproximar haciendo :

$$\frac{\partial f}{\partial u_k}(u) \approx \left(f\begin{pmatrix} u_0 \\ .. \\ u_k + h \\ .. \\ u_{N-1} \end{pmatrix} - f\begin{pmatrix} u_0 \\ .. \\ u_k \\ .. \\ u_{N-1} \end{pmatrix} \right) /h. \tag{6.124}$$

Para expresar la aproximación del gradiente de forma más compacta definimos la base canónica dada por:

$$e^k = {}^t(0, .., 0, \underbrace{1}_{k}, 0, .., 0). \tag{6.125}$$

Podemos expresar entonces la aproximación de la matriz gradiente como :

$$\nabla f(u) \cong \left(\frac{f(u + h_0 \cdot e^0) - f(u)}{h_0}, ..., \frac{f(u + h_{N-1} \cdot e^{N-1}) - f(u)}{h_{N-1}}\right), \tag{6.126}$$

El paso h_k puede ser el mismo para todos y un parámetro de la función o calcularse automáticamente para cada k usando la precisión de la aritmética t usando la fórmula:

$$h_k = |u_k|\sqrt{\frac{1}{2^t}} + \epsilon. \tag{6.127}$$

Por tanto, el algoritmo completo para calcular la matriz derivada quedaría:

Algoritmo cálculo matriz derivada en el vector u
hacemos una copia del vector u y lo guardamos en v
Lanzamos un proceso iterativo donde en cada iteración:
sumamos h a la componente k del vector v
calculamos la derivada parcial (que es un vector)
dicha derivada parcial se pone en la columna k de la matriz $\nabla f(u)$
actualizamos v para que de nuevo sea igual a u.
fin iteraciones

Finalmente el algoritmo completo de Newton-Raphson se puede describir como:

Algoritmo Método de Newton-Raphson para sistemas
Lanzamos un proceso iterativo donde en cada iteración:
Calculamos matriz derivada $\nabla f(u)$
Resolvemos el sistema lineal $\nabla f(u) \cdot x = -f(u)$
se actualiza u=u+x;
fin iteraciones
Los criterios de salida de las iteraciones son que $f(u)$ sea igual a cero o que la distancia entre u y su versión anterior sea menor que TOL.

6.6. Optimización

6.6.1. Introducción

En muchos ámbitos de conocimiento se utilizan modelos matemáticos para simular la realidad y tomar decisiones. Formalmente podemos decir que la optimización consiste en buscar la mejor solución a un problema dado de acuerdo con un criterio. Normalmente este criterio consiste en minimizar una función objetivo respecto a unos parámetros. Para introducir de manera concreta esta idea vamos usar el modelo de regresión lineal visto en el tema de interpolación. Allí se explicó que la regresión lineal consiste en aproximar una nube de puntos (x_i, y_i) a través de una recta $y = ax + b$ minimizando el error cuadrático:

$$E(a,b) = \sum_i (ax_i + b - y_i)^2, \tag{6.128}$$

por tanto, la obtención de la recta de regresión la podemos interpretar como un problema de optimización donde se trata de buscar los mejores valores de a y b a través de la minimización del criterio dado por la ecuación anterior

Vamos a desarrollar ahora un ejemplo en el ámbito de la economía usando el modelo de Cobb–Douglas que simula la productividad de una empresa utilizando el siguiente modelo:

$$P = d \cdot K^a \cdot L^b \cdot M^c \tag{6.129}$$

donde P representa el valor en el mercado de la producción de la empresa en un periodo, K representa el capital fijo invertido en la empresa (equipamiento, edificios, etc..), L representa el coste de personal y M los costes de los suministros necesarios para la producción, los números a, b, c y d son parámetros que hay que ajustar a cada caso en concreto. El primer problema de optimización que se plantea en este contexto es, dada una empresa, como ajustar los parámetros a, b, c, d. Este ajuste se suele realizar a partir de datos conocidos de la empresa en los últimos años. Es decir, supongamos que en los $N_{años}$ últimos años conocemos los datos $(P_i, K_i, L_i.M_i)$ con $i = 1, .., N_{años}$. Podemos ajustar a, b, c, d utilizando estos valores buscando el mínimo del error cuadrático:

$$E(a, b, c, d) = \sum_{i=1}^{N_{años}} \left(P_i - d \cdot K_i^a \cdot L_i^b \cdot M_i^c \right)^2 \tag{6.130}$$

esto se denomina un problema de optimización que se resuelve buscando el mínimo de la anterior función. Una vez calculados los valores de a, b, c y d, el siguiente problema que se plantea es como usar el modelo para tomar decisiones, por ejemplo, decidir cual es el balance entre las cantidades K, L y M que optimiza el beneficio de la empresa. Dicho beneficio viene dado por

$$B(K, L, M) = d \cdot K^a \cdot L^b \cdot M^c - L - M - rK \tag{6.131}$$

donde r es la tasa de coste de mantenimiento de los bienes necesarios para la producción (es decir, intereses deuda capital invertido, mantenimiento equipamiento, edificios, etc..). Esto plantea un segundo problema de optimización que consiste en maximizar la anterior función beneficio. Nótese que buscar el máximo de la función $B(K, L, M)$ es lo mismo que buscar el mínimo de la función $-B(K, L, M)$ donde simplemente le hemos cambiado el signo. Para optimizar el beneficio tenemos que añadir alguna restricción, pues en caso contrario, con este modelo, cuanto más se invierta, más beneficio habrá y la optimización nos llevaría a ir subiendo sin límites todo el capital circulante $(K + L + M)$. Para resolver esto, podemos añadir la restricción de que todo el capital circulante se ajuste a una cantidad prefijada, C, en este caso, la optimización nos daría como distribuir el capital C entre los diferentes apartados K, L y M para que el beneficio sea máximo. Esta restricción se puede añadir de varias maneras. Nosotros aquí elegiremos una forma muy sencilla que consiste en penalizar en la función a optimizar que no se cumpla la restricción. Para ello la función que vamos a minimizar para optimizar el beneficio será:

$$F(K, L, M) = (K + L + M - C)^2 - B(K, L, M) \tag{6.132}$$

Nótese que le cambiamos el signo al beneficio $B(K, L, M)$ para optimizarlo buscando mínimos y no máximos.

6.6.2. Aplicación del método de Newton-Raphson a la optimización

Desde un punto de vista matemático un problema de optimización consiste en la búsqueda del mínimo de una función $F(u) : R^N \to R$, donde $u = {}^t(u_0, ..., u_{N-1})$.

La optimización está muy relacionada con el problema de búsqueda de ceros de la función derivada $f(u) = \nabla F(u) : R^N \to R^N$, puesto que si u es un mínimo local de una función, entonces $f(u) = \nabla F(u) = 0$. Por tanto, utilizando el método de Newton-Raphson para sistemas explicado en la sección anterior, podemos plantear el siguiente algoritmo de optimización:

Algoritmo de optimización de Newton-Raphson
Dada una función $F(u)$ y una aproximación inicial del mínimo u:

(1) Se construye la función derivada $f(u) = \nabla F(u)$ analíticamente, o de forma discreta haciendo:

$$f(u) = \left(\frac{F(u + h_0 \cdot e^0) - F(u)}{h_0}, .., \frac{F(u + h_{N-1} \cdot e^{N-1}) - F(u)}{h_{N-1}} \right)$$

donde e^k está definido en (6.125) y h_k puede ser constante o definido como en (6.127).

(2) Se aplica a la función $f(u)$ el método de Newton-Raphson para sistemas descrito en la sección anterior.

6.6.3. Método de gradiente descendente atenuado

Dado que estamos intentando minimizar una función $F(u)$, como la dirección de máximo descenso de la función es la dirección opuesta al gradiente (es decir $-\nabla F(u)$), podemos, a partir de una aproximación inicial, ir mejorandola moviendonos en la dirección de maximo descenso pero preservando que la función $F(u)$ vaya disminuyendo en cada iteración, es decir, en cada iteración del proceso hacemos

$$u^{k+1} = u^k - \lambda_k \nabla F(u^k) \tag{6.133}$$

donde $\lambda_k > 0$ debe ajustarse para que $F(u^{k+1}) < F(u^k)$. Para ello se tiene en cuenta que si $\lambda_k > 0$ es suficientemente pequeño y u^k no es el mínimo, se debe cumplir que $F(u^{k+1}) < F(u^k)$. Ello da lugar al siguiente método denominado de gradiente descendente atenuado:

Algoritmo de optimización de gradiente descendente atenuado
Partimos de una función $F(u)$, una aproximación inicial del mínimo u y un valor inicial $\lambda > 0$.
Lanzamos un proceso iterativo. En cada iteración hacemos:
Calcular $F(u)$ y $\nabla F(u)$
Calcular $v = u - \lambda \nabla F(u)$ y $F(v)$.
Si $(F(v) \geq F(u))$ no aceptamos v como nueva solución, disminuimos el valor de λ y volvemos a calcular v y volvemos a comprobar si $F(v) \geq F(u)$.
Si $F(v) < F(u)$ aceptamos v como nueva solución y aumentamos el valor de λ.
Fin Iteraciones
Los criterios de parada de las iteraciones son que u o $F(u)$ estén muy próximos a su valor en la iteración anterior.

el calificativo de atenuado viene de que "atenuamos" en cada iteración el valor de λ para que el valor de la función objetivo descienda. Otra posibilidad sería tomar un λ fijo para todas las iteraciones y no exigir que $F(u^{k+1}) < F(u^k)$, en cuyo caso el método de gradiente descendente no sería "atenuado".

6.6.4. Método de Newton-Raphson atenuado

El método de Newton-Raphson tiene la desventaja de que no tiene en cuenta que el valor de la función objetivo disminuya a lo largo de las iteraciones. Para evitar eso vamos a introducir un parámetro de atenuación λ que nos servirá para combinar el método de Newton-Raphson y el método de gradiente descendente. En primer lugar recordamos que con el método de Newton-Raphson, para pasar de la iteración u^n a la iteración u^{n+1} hacemos $u^{n+1} = u^n + x$ donde x es la solución del sistema:

$$D^2F(u^n)x = -\nabla F(u^n). \tag{6.134}$$

Por otro lado, en el método de gradiente descendente atenuado, $u^{n+1} = u^n + x$ donde x cumple:

$$I_d x = -\lambda \nabla F(u^n). \tag{6.135}$$

donde I_d es la matriz identidad. El método de Newton-Raphson atenuado consiste en combinar ambos métodos haciendo $u^{n+1} = u^n + x$ donde x es la solución del sistema:

$$(I_d + \lambda D^2F(u^n))x = -\lambda \nabla F(u^n). \tag{6.136}$$

nótese que, en la anterior ecuación, si λ es grande, la parte de I_d es despreciable y la solución del método se aproxima al método de Newton-Raphson y si λ es pequeño entonces la parte de $\lambda D^2F(u^n)$ es despreciable y la solución del método se aproxima al método de gradiente descendente. En la práctica λ se toma lo más grande posible siempre que se cumpla que $F(u^{n+1}) < F(u^n)$. Es decir nos intentamos aproximar lo más posible al método de Newton-Raphson que es más eficiente que el método de

gradiente descendente cuando nos vamos acercando al mínimo de la función objetivo. Ello nos lleva al siguiente algoritmo:

Algoritmo de optimización de Newton-Raphson atenuado
Partimos de una función $F(u)$, una aproximación inicial del mínimo u y un valor inicial $\lambda > 0$.
Lanzamos un proceso iterativo. En cada iteración hacemos:
Calculamos $\nabla F(u)$ y la matriz de derivada segunda $D^2F(u)$
Resolvemos el sistema $(I_d + \lambda D^2F(u))x = -\lambda \nabla F(u)$
Si ($F(u+x) < F(u)$) aceptamos $u+x$ como nueva solución e incrementamos el valor de λ.
Si ($F(u+x) \geq F(u)$) no aceptamos $u+x$ como nueva solución y disminuimos el valor de λ.
Fin iteraciones Los criterios de parada de las iteraciones son que $\nabla F(u)$ sea cero, o que u esté muy próximo a su valor en la iteración anterior.

6.7. Problemas resueltos

Problema 65 *Tomar $N = 2$ y demostrar que la norma $\| x \|_2$ verifica las propiedades de la definición de norma*

$$\|x\|_2 = \sqrt[2]{|x_1|^2 + |x_2|^2}.$$

Solución: Las propiedades que debe verificar, para cumplir con la definición de norma, son:

1. $\|x\|_2 = 0 \Longleftrightarrow x = 0 \quad \rightarrow \quad \sqrt[2]{|x_1|^2 + |x_2|^2} = 0 \Longleftrightarrow (x_1, x_2) = (0,0)$,

2. $\|\lambda x\|_2 = |\lambda| \, \|x\|_2 , \forall \lambda \in K \;\; y \;\; x \in E$;

$$\|\lambda x\|_2 = \sqrt[2]{|\lambda x_1|^2 + |\lambda x_2|^2} = \sqrt[2]{|\lambda|^2 \left(|x_1|^2 + |x_2|^2 \right)} = |\lambda| \sqrt[2]{|x_1|^2 + |x_2|^2} = |\lambda| \, \|x\|_2 .$$

3. $\|x+y\|_2 \leq \|x\|_2 + \|y\|_2 , \forall x, y \in E$;

$$\|x+y\|_2 \leq \|x\|_2 + \|y\|_2 \Longleftrightarrow$$

$$\Longleftrightarrow |x_1 + y_1|^2 + |x_2 + y_2|^2 \leq \left(\sqrt[2]{|x_1|^2 + |x_2|^2} + \sqrt[2]{|y_1|^2 + |y_2|^2} \right)^2 \Longleftrightarrow$$

$$\Longleftrightarrow x_1^2 + 2x_1y_1 + y_1^2 + x_2^2 + 2x_2y_2 + y_2^2 \leq x_1^2 + x_2^2 + 2\sqrt{x_1^2 + x_2^2}\sqrt{y_1^2 + y_2^2} + y_1^2 + y_2^2$$

$$\iff x_1y_1+x_2y_2 \leq \sqrt{x_1^2+x_2^2}\sqrt{y_1^2+y_2^2} \iff (x_1y_1+x_2y_2)^2 \leq \left(x_1^2+x_2^2\right)\left(y_1^2+y_2^2\right)$$
$$\iff 2x_1y_1x_2y_2 \leq x_1^2y_2^2+x_2^2y_1^2 \iff 0 \leq (x_1y_2+x_2y_1)^2,$$

que siempre es cierto, con lo que queda demostrado.

Problema 66 *Demostrar que*

$$Lim_{p\to\infty} \|x\|_p = max_i |x_i| .$$

Solución:

$Lim_{p\to\infty} \|x\|_p = Lim_{p\to\infty} \sqrt[p]{\sum_{i=1}^N |x_i|^p}$.

Extraemos el máximo componente de x, x_{max}.

$$Lim_{p\to\infty} \sqrt[p]{\sum_{i=1}^N |x_i|^p} = Lim_{p\to\infty} \sqrt[p]{|x_{max}|^p \sum_{i=1}^N \frac{|x_i|}{|x_{max}|}^p} =$$

$$= |x_{max}|\, Lim_{p\to\infty} \left(\sum_{i=1}^N \left(\frac{|x_i|}{|x_{max}|}\right)^p\right)^{1/p}.$$

Todos los elementos $\frac{|x_i|}{|x_{max}|}$ son menores o iguales que 1, con lo que

$$Lim_{p\to\infty} \left(\frac{|x_i|}{|x_{max}|}\right)^p = \begin{cases} 1 & si \quad |x_i| = |x_{max}| \\ 0 & si \quad |x_i| < |x_{max}| \end{cases},$$

entonces

$$|x_{max}|\, Lim_{p\to\infty} \left(\sum_{i=1}^N \left(\frac{|x_i|}{|x_{max}|}\right)^p\right)^{1/p} =$$
$$= |x_{max}|\, Lim_{p\to\infty} (0+\ldots+0+1+\ldots+1)^{1/p} = |x_{max}|, \text{ c.q.d.}$$

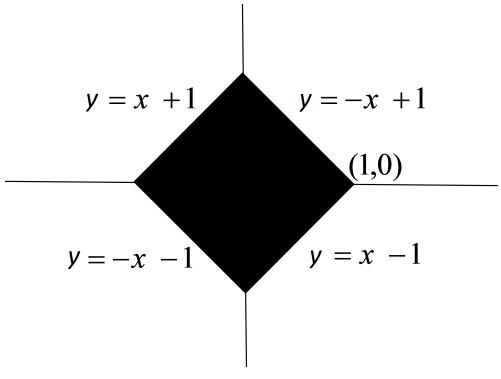
Problema 67 *Tomar $N = 2$, y dibujar el lugar geométrico de los vectores $x = (x_1, x_2)$ que verifican*

1. $\|x\|_1 < 1$,
2. $\|x\|_2 < 1$,
3. $\|x\|_\infty < 1$.

Solución:

1. $\|x\|_1 < 1 \Longrightarrow |x| + |y| < 1$.

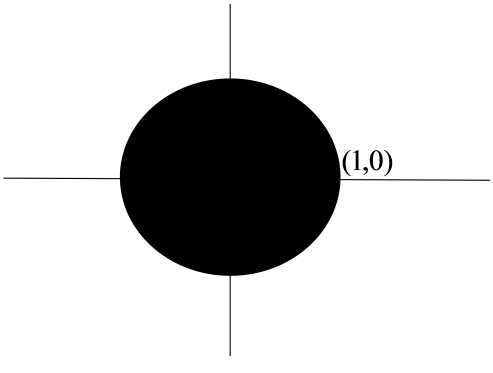
   El borde del lugar geométrico representa, en cada cuadrante, una recta distinta, por ejemplo en el primer cuadrante es la recta $y < 1 - x$. Reuniendo la información de los 4 cuadrantes sale la siguiente figura:



Lugar geométrico de $\|x\|_1 < 1$

2. $\|x\|_2 < 1 \Longrightarrow \sqrt{x^2 + y^2} < 1 \Longrightarrow x^2 + y^2 < 1$

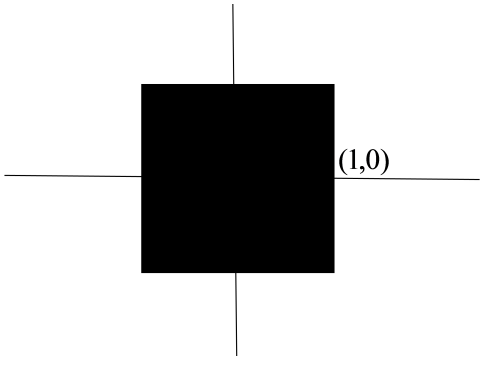
   Esta es la ecuación de un círculo de radio menor que 1 y centro el origen.



Lugar geométrico de $\|x\|_2 < 1$

3. $\|x\|_\infty < 1 \Longrightarrow max(|x|, |y|) < 1$

   Si $max(|x|, |y|) < 1$ entonces $|x| < 1$ y $|y| < 1$, por tanto su lugar geométrico es:



Lugar geométrico de $\|x\|_\infty < 1$

Problema 68 *Tomar $N = 2$ y demostrar la siguiente desigualdad*

$$\| x \|_\infty \leq \| x \|_2 \leq \| x \|_1 .$$

Solución: Esta desigualdad es equivalente a lo siguiente:

$$\max(|x_1|, |x_2|) \leq \sqrt{x_1^2 + x_2^2} \leq |x_1| + |x_2|,$$

1. $\max(|x_1|,|x_2|) \leq \sqrt{x_1^2+x_2^2} \iff |x_{max}| \leq \sqrt{x_1^2+x_2^2} \iff x_{max}^2 \leq x_1^2+x_2^2$
   Esta desigualdad siempre es cierta ya que x_{max} es o bien x_1 o bien x_2.

2. $\sqrt{x_1^2+x_2^2} \leq |x_1|+|x_2| \iff x_1^2+x_2^2 \leq (|x_1| + |x_2|)^2 \iff x_1^2+x_2^2 \leq |x_1|^2+2\,|x_1|\,|x_2|+$

   $|x_2|^2 \iff x_1^2 + x_2^2 \leq x_1^2 + 2\,|x_1|\,|x_2| + x_2^2 \iff 0 \leq 2\,|x_1|\,|x_2|$

   Esto siempre es cierto ya que el producto de valores positivos siempre es positivo (o igual a cero si algún x_i es cero).

3. $\max(|x_1|, |x_2|) \leq |x_1| + |x_2|$. Es trivial.

De estas demostraciones se deduce que las distintas normas coinciden cuando el vector x está sobre uno de los ejes de coordenadas.

Problema 69 *Demostrar que si A, B son dos matrices de dimensión $N \times N$, entonces para cualquier norma de matrices subordinada a una norma vectorial se verifica*

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

Solución:

$$\sup_x \frac{\|ABx\|}{\|x\|} = \sup_x \frac{\|ABx\|}{\|x\|}\frac{\|Bx\|}{\|Bx\|} \leq \sup_x \frac{\|Bx\|}{\|x\|} \cdot \sup_x \frac{\|ABx\|}{\|Bx\|} = \|B\| \cdot \|A\|,$$

Problema 70 *Demostrar que los autovalores de A son los ceros del polinomio característico $P(\lambda)$.*

Solución: Dado un autovalor λ_i de una matriz A y su correspondiente autovector x_i se verifica:

$$Ax_i = \lambda_i x_i \Longrightarrow (A - \lambda_i Id)x_i = 0,$$

como $x_i \neq 0$, entonces

$$|A - \lambda_i Id| = 0 \Longrightarrow P(\lambda) = 0, \text{ c.q.d.}$$

Problema 71 *Calcular los autovectores de la matriz*

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix},$$

y determinar una base ortonormal de R^3 de autovectores de A.

Solución: Calculamos los autovalores de A :

$$|A - \lambda_i Id| = \begin{vmatrix} 1-\lambda & 1 & 0 \\ 1 & 1-\lambda & 0 \\ 0 & 0 & 2-\lambda \end{vmatrix} = ((1-\lambda)^2 - 1)(2-\lambda) = 0.$$

que tiene como raíces $\lambda_1 = 0$, $\lambda_2 = 2$, $\lambda_3 = 2$. Calculamos a continuación los autovectores de A :

$$\lambda_1 = 0 \quad \rightarrow \quad \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

$$\begin{cases} x_1 + x_2 = 0 \\ x_1 + x_2 = 0 \\ 2x_3 = 0 \end{cases} \begin{cases} x_1 = -x_2 \\ \\ x_3 = 0 \end{cases} \rightarrow \quad \bar{x}_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \\ 0 \end{pmatrix},$$

$$\lambda_2, \lambda_3 = 2 \quad \rightarrow \quad \begin{pmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

$$\begin{cases} -x_1 + x_2 = 0 \\ x_1 - x_2 = 0 \\ x_3 \text{ libre} \end{cases} \quad \begin{cases} x_1 = x_2 \\ \\ x_3 \text{ libre} \end{cases} \quad \rightarrow \quad \bar{x}_2 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}, \bar{x}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Por tanto la matriz

$$B = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

contiene los autovectores de A que forman una base ortonormal en R^3.

Problema 72 *Calcular a, b, c para que los vectores columna de la siguiente matriz formen una base ortogonal de vectores.*

$$\begin{pmatrix} 2 & 1 & b \\ a & 1 & -3 \\ -1 & 2 & c \end{pmatrix},$$

Solución: para que los vectores columna de la matriz formen una base de vectores perpendiculares debe cumplirse :

$$2 \cdot 1 + a \cdot 1 - 1 \cdot 2 = 0 \quad \rightarrow a = 0,$$
$$2 \cdot b - c = 0,$$
$$b - 3 + 2c = 0,$$

resolviendo el sistema para calcular b, c obtenemos $b = \frac{3}{5}$ y $c = \frac{6}{5}$.

Problema 73 *Calcular las normas 2, 1 e infinito de la matriz*

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}.$$

Solución: para calcular $\| A \|_2$, calculamos primero los autovalores de la matriz

$${}^tAA = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 2 & 4 \end{pmatrix},$$

que se calculan usando el polinomio característico

$$\begin{vmatrix} 2-\lambda & 2 \\ 2 & 4-\lambda \end{vmatrix} = \lambda^2 - 6\lambda + 4 = 0,$$

cuyas raíces son $3 \pm \sqrt{5}$. Por tanto:

- $\| A \|_2 = \sqrt{\rho\,({}^tAA)} = \sqrt{3+\sqrt{5}}$.
- $\| A \|_1 = max_j\,(\sum_i \mid a_{ij} \mid) = 2$.
- $\| A \|_\infty = max_i \left(\sum_j \mid a_{ij} \mid\right) = 3$.

Problema 74 *Demostrar la siguiente igualdad:*

$$\rho({}^tAA) = \rho(A\,{}^tA).$$

Solución: Veamos que los polinomios característicos coinciden :

$$\left|{}^tAA - \lambda_i Id\right| = \left|{}^tA\right|^{-1}\left|{}^tAA - \lambda_i Id\right|\left|{}^tA\right| = \left|A\,{}^tA - \lambda_i\left({}^tA\right)^{-1}{}^tA\right| = \left|A\,{}^tA - \lambda_i Id\right|.$$

Problema 75 *Demostrar que si los autovectores de una matriz A de dimensión $N \times N$ forman una base ortonormal de R^N, entonces para la norma 2 se cumple:*

$$\chi(A) = \|A\|_2 \cdot \left\|A^{-1}\right\|_2 = \frac{max_i\{|\lambda_i|\}}{min_i\{|\lambda_i|\}}.$$

Solución: Al ser una base de autovectores ortonormal, la norma $\|A\|_2 = \rho(A) = max_i\,\{|\lambda_i|\}$

Los autovalores de A^{-1} vienen dados por:

$$Ax_i = \lambda_i x_i \Longrightarrow A^{-1}Ax_i = A^{-1}\lambda_i x_i \Longrightarrow \frac{1}{\lambda_i}x_i = A^{-1}x_i \Longrightarrow A^{-1}x_i = \lambda_i' x_i,$$

donde $\lambda_i' = \frac{1}{\lambda_i}$, es decir, los autovalores de A^{-1} son los inversos de los de A y sus autovectores son los mismos, luego la norma de $\left\|A^{-1}\right\|_2 = \rho(A^{-1})$

$$\left\|A^{-1}\right\|_2 = max_i\left\{\left|\lambda_i'\right|\right\} = max_i\left\{\left|\frac{1}{\lambda_i}\right|\right\} = \frac{1}{min_i\,\{|\lambda_i|\}},$$

entonces

$$\chi(A) = \|A\|_2 \cdot \left\|A^{-1}\right\|_2 = max_i\,\{|\lambda_i|\} \cdot \frac{1}{min_i\,\{|\lambda_i|\}} = \frac{max_i\{|\,\lambda_i\,|\}}{min_i\{|\,\lambda_i\,|\}}.$$

Problema 76 *Calcular el condicionamiento de la siguiente matriz para las normas 1, 2 e ∞.*

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}.$$

Solución:

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \qquad \Rightarrow \qquad A^{-1} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix},$$

$$\chi(A)_2 = \|A\|_2 \|A^{-1}\|_2 = \sqrt{\rho({}^tAA)}\sqrt{\rho({}^tA^{-1}A^{-1})} = 2.618,$$

$$\chi(A)_1 = \|A\|_1 \|A^{-1}\|_1 = 2 \cdot \frac{3}{2} = 3,$$

$$\chi(A)_\infty = \|A\|_\infty \|A^{-1}\|_\infty = 3 \cdot 1 = 3.$$

Problema 77 *Calcular el condicionamiento para la norma* 2*, de las siguientes matrices:*

1. $A = \begin{pmatrix} 2 & 2 & -2 \\ 2 & 1 & 1 \\ -2 & 1 & 1 \end{pmatrix}$.

2. $A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$.

Solución: El condicionamiento para la norma 2 de una matriz es $\chi(A) = \|A\|_2 \cdot \left\|A^{-1}\right\|_2$. Primero calculamos los autovalores de A:

1. $\begin{vmatrix} 2-\lambda & 2 & -2 \\ 2 & 1-\lambda & 1 \\ -2 & 1 & 1-\lambda \end{vmatrix} = (2-\lambda)\left(\lambda^2 - 2\lambda + 8\right)$,

   de donde obtenemos

   $$\lambda_1 = 2, \quad \lambda_2 = -2, \lambda_3 = 4.$$

   Esta matriz es simétrica, luego posee una base ortonormal de autovectores, con lo que el condicionamiento de A se puede calcular como:

   $$\chi(A) = \|A\|_2 \cdot \left\|A^{-1}\right\|_2 = \frac{max_i\{|\ \lambda_i\ |\}}{min_i\{|\ \lambda_i\ |\}} = \frac{4}{2} = 2.$$

2. $\begin{vmatrix} 2-\lambda & -1 & 0 \\ -1 & 2-\lambda & -1 \\ 0 & -1 & 2-\lambda \end{vmatrix} = 4 - 10\lambda + 6\lambda^2 - \lambda^3 = 0$

   de donde

   $$\lambda_1 = 2 \quad \lambda_2 = 2 + \sqrt{2} \quad \lambda_3 = 2 - \sqrt{2}.$$

   Esta matriz también es una matriz simétrica, con lo que sus autovectores forman una base ortonormal y su condicionamiento es:

   $$\chi(A) = \|A\|_2 \cdot \left\|A^{-1}\right\|_2 = \frac{max_i\{|\ \lambda_i\ |\}}{min_i\{|\ \lambda_i\ |\}} = \frac{2+\sqrt{2}}{2-\sqrt{2}} = 3 + 2\sqrt{2}.$$

Problema 78 *Sean las matrices A, R. Demostrar que la matriz A, y la matriz $B = R^{-1}AR$ poseen los mismos autovalores*

Solución:

$$Bx_i = \lambda_i x_i \Longrightarrow \left(R^{-1}AR\right) x_i = \lambda_i x_i \Longrightarrow RR^{-1}ARx_i = R\lambda_i x_i \Longrightarrow$$

$$\Longrightarrow ARx_i = \lambda_i Rx_i \Longrightarrow Ay_i = \lambda_i y_i,$$

de donde se deduce que los autovalores son los mismos y los autovectores están relacionados por la siguiente igualdad: $y_i = Rx_i$, c.q.d.

Problema 79 *Se considera la matriz*

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},$$

calcular el ángulo α tal que la matriz

$$R = \begin{pmatrix} cos(\alpha) & sen(\alpha) \\ -sen(\alpha) & cos(\alpha) \end{pmatrix},$$

verifique que la matriz $B = R^{-1}AR$ sea diagonal.

Solución: Realizamos el cálculo de la matriz B :

$$B = R^{-1}AR = \begin{pmatrix} cos(\alpha) & -sen(\alpha) \\ sen(\alpha) & cos(\alpha) \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} cos(\alpha) & sen(\alpha) \\ -sen(\alpha) & cos(\alpha) \end{pmatrix} =$$

$$= \begin{pmatrix} -2cos(\alpha)sen(\alpha) + 1 & 2cos^2\alpha - 1 \\ 2cos^2\alpha - 1 & 2cos(\alpha)sen(\alpha) + 1 \end{pmatrix} = \begin{pmatrix} b_1 & 0 \\ 0 & b_2 \end{pmatrix}.$$

Se tiene que cumplir que los elementos que están fuera de la diagonal sean iguales a cero, es decir

$$2cos^2\alpha - 1 = 0 \quad \rightarrow \quad cos(\alpha) = \pm\sqrt{\frac{1}{2}}.$$

De esta igualdad se obtiene los posibles valores del ángulo α : $\alpha = \frac{\pi}{4}, \alpha = \frac{3\pi}{4}$. Las posibles matrices de rotación quedan como sigue:

$$R_1 = \begin{pmatrix} cos(\frac{\pi}{4}) & sen(\frac{\pi}{4}) \\ -sen(\frac{\pi}{4}) & cos(\frac{\pi}{4}) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \\ -\frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{pmatrix},$$

$$R_2 = \begin{pmatrix} cos(\frac{3\pi}{4}) & sen(\frac{3\pi}{4}) \\ -sen(\frac{3\pi}{4}) & cos(\frac{3\pi}{4}) \end{pmatrix} = \begin{pmatrix} -\frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \\ -\frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} \end{pmatrix}.$$

Calculamos los elementos de la diagonal usando las fórmulas

$$b_1 = -2cos(\alpha)sen(\alpha) + 1, \quad b_2 = 2cos(\alpha)sen(\alpha) + 1.$$

obteniendo finalmente las siguientes soluciones posibles:

$$B_1 = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}.$$

Problema 80 *Sabiendo que*

$$\tan(2\alpha) = x,$$

calcular el $sen(\alpha)$ *y* $cos(\alpha)$ *a partir del valor de* x *sin calcular explícitamente* α *ni llamar a ninguna función trigonométrica*

Solución: Utilizaremos las relaciones trigonométricas

$$\begin{aligned} \tan(\alpha) &= -\cot(2\alpha) + sign(\cot(2\alpha))\sqrt{1+\cot^2(2\alpha)}, \\ cos(\alpha) &= \frac{1}{\sqrt{1+\tan^2(\alpha)}}, \\ sen(\alpha) &= \tan(\alpha)cos(\alpha). \end{aligned}$$

donde $sign(x) = 1$ si $x \geq 0$ y $sign(x) = -1$ si $x < 0$. Siguiendo estas fórmulas obtenemos :

$$\begin{aligned} \tan(\alpha) &= -\frac{1}{x} + sign(\frac{1}{x})\sqrt{1+\frac{1}{x^2}}, \\ cos(\alpha) &= \frac{1}{\sqrt{1+\left(-\frac{1}{x} + sign(\frac{1}{x})\sqrt{1+\frac{1}{x^2}}\right)^2}}, \\ sen(\alpha) &= \frac{-\frac{1}{x} + sign(\frac{1}{x})\sqrt{1+\frac{1}{x^2}}}{\sqrt{1+\left(-\frac{1}{x} + sign(\frac{1}{x})\sqrt{1+\frac{1}{x^2}}\right)^2}}. \end{aligned}$$

usando los resultados de este problema podemos acelerar el cálculo del $cos(\alpha)$ y $sen(\alpha)$ en el método de Jacobi para calcular los autovalores y autovectores de matrices simétricas.

Problema 81 *Utilizar el método de Jacobi para aproximar los autovalores y autovectores de la matriz:*

$$A = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Solución:

$$R(\alpha) = \begin{pmatrix} cos(\alpha) & 0 & sen(\alpha) \\ 0 & 1 & 0 \\ -sen(\alpha) & 0 & cos(\alpha) \end{pmatrix},$$

$$\tan(2\alpha) = \frac{2a_{pq}}{(a_{qq} - a_{pp})} = \frac{2}{(1-2)} = -2 \quad \rightarrow \quad \alpha = \frac{\arctan(-2)}{2} = -0.55357,$$

por tanto

$$R(\alpha) = \begin{pmatrix} cos(\alpha) & 0 & sen(\alpha) \\ 0 & 1 & 0 \\ -sen(\alpha) & 0 & cos(\alpha) \end{pmatrix} = \begin{pmatrix} 0.85065 & 0 & -0.52573 \\ 0 & 1 & 0 \\ 0.52573 & 0 & 0.85065 \end{pmatrix},$$

y al actualizar A queda

$$A = R^{-1}AR = \begin{pmatrix} 2.618 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.38197 \end{pmatrix},$$

que es una matriz diagonal. Por tanto, los autovalores de A son los elementos de la diagonal $(2.618, 1, 0.38197)$. Como en este caso, hay una única matriz de rotación al conseguir transformar A en una matriz diagonal en una iteración, tendremos que los autovectores de A son simplemente los vectores columnas de $R(\alpha)$. Es decir el autovector del autovalor 2.618 es ${}^t(0.85065,\ 0,\ 0.52573)$, el autovector del autovalor 1 es ${}^t(0, 1, 0)$, y el autovector del autovalor 0.38197 es ${}^t(-0.52573,\ 0,\ 0.85065)$.

Problema 82 *Aplicar el método de la potencia para aproximar el autovalor máximo, y el autovector asociado, de las siguientes matrices, dando* 3 *pasos en el método, hasta calcular* u^3 *y partiendo de* $u^0 = {}^t(1, 1)$.

$$A = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix},$$
$$A = \begin{pmatrix} -3 & 0 \\ 1 & 1 \end{pmatrix}.$$

Solución: En este problema vamos a utilizar la norma Euclídea aunque cualquier otra norma también sería válida. La norma infinito, por ejemplo, simplificaría los cálculos ya que es inmediato obtener el máximo de un vector.

1. $A = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$,

$$u^1 = A\frac{u^0}{\|u^0\|} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{3}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} \end{pmatrix} \quad \rightarrow \|u^1\| = \sqrt{5} = 2.2361,$$

$$u^2 = A\frac{u^1}{\|u^1\|} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \frac{3}{2\sqrt{5}}\sqrt{2} \\ \frac{1}{2\sqrt{5}}\sqrt{2} \end{pmatrix} = \begin{pmatrix} \frac{7}{10}\sqrt{5}\sqrt{2} \\ \frac{1}{10}\sqrt{5}\sqrt{2} \end{pmatrix} \rightarrow \|u^2\| = \sqrt{5} = 2.2361,$$

$$u^3 = A\frac{u^2}{\|u^2\|} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \frac{7}{10}\sqrt{2} \\ \frac{1}{10}\sqrt{2} \end{pmatrix} \rightarrow \|u^3\| = \frac{1}{5}\sqrt{113} = 2.126.$$

El autovalor máximo aproximado es $\lambda_{max} = 2.126$ y su autovector asociado es:

$$x_{\lambda_{max}} = \frac{u^3}{\|u^3\|} = \frac{1}{2.126}\begin{pmatrix} \frac{3}{2}\sqrt{2} \\ \frac{1}{10}\sqrt{2} \end{pmatrix} = \begin{pmatrix} 0.99779 \\ 0.066519 \end{pmatrix}$$

2. $A = \begin{pmatrix} -3 & 0 \\ 1 & 1 \end{pmatrix}$,

$$u^1 = A\frac{u^0}{\|u^0\|} = \begin{pmatrix} -3 & 0 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} -\frac{3}{2}\sqrt{2} \\ \sqrt{2} \end{pmatrix} \rightarrow \|u^1\| = \frac{1}{2}\sqrt{26} = 2.5495,$$

$$u^2 = A\frac{u^1}{\|u^1\|} = \begin{pmatrix} -3 & 0 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} -\frac{3\sqrt{52}}{26} \\ \frac{\sqrt{52}}{13} \end{pmatrix} = \begin{pmatrix} \frac{9\sqrt{52}}{26} \\ -\frac{\sqrt{52}}{26} \end{pmatrix} \rightarrow \|u^2\| = 2.5115,$$

$$u^3 = A\frac{u^2}{\|u^2\|} = \begin{pmatrix} -3 & 0 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} \frac{9\sqrt{52}\sqrt{1066}}{2132} \\ -\frac{\sqrt{52}\sqrt{1066}}{2132} \end{pmatrix} = \begin{pmatrix} -\frac{27\sqrt{52}\sqrt{1066}}{2132} \\ \frac{2\sqrt{52}\sqrt{1066}}{533} \end{pmatrix} \rightarrow \|u^3\| = 3.1098.$$

Dado que hay alternancia de signo en los coeficientes de los vectores u^2 y u^3, su producto escalar es negativo y por tanto el autovalor máximo aproximado es negativo y viene dado por

$$\lambda_{max} = -3.1098,$$

su autovector asociado es

$$x_{\lambda_{max}} = \frac{u^3}{\|u^3\|} = \frac{1}{3.1098}\begin{pmatrix} -\frac{27\sqrt{52}\sqrt{1066}}{2132} \\ \frac{2\sqrt{52}\sqrt{1066}}{533} \end{pmatrix} = \begin{pmatrix} -0.9588 \\ 0.28409 \end{pmatrix},$$

Problema 83 *Calcular el autovalor mayor y el autovector correspondiente de la matriz*

$$\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix},$$

utilizando el método de la potencia, dando 2 iteraciones del método a partir de $u^0 = {}^t(1,1)$ *y tomando como norma* $\|u\| = max_i\,|u_i|$

Solución:

$$\|u^0\| = 1 \rightarrow u^1 = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$\left\|u^1\right\| = 1 \to u^2 = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}.$$

El producto escalar $(u^1, u^2) = 2 > 0$ y por tanto el autovalor máximo es $\left\|u^2\right\| = 2$ y el autovector asociado normalizado es $\frac{u^2}{\|u^2\|} = \begin{pmatrix} 1 \\ -1/2 \end{pmatrix}$.

Problema 84 *Utilizar el método de la potencia inversa para aproximar el autovalor más pequeño de la matriz*

$$A = \begin{pmatrix} -2 & 1 \\ 0 & 3 \end{pmatrix},$$

llegar hasta u^2 *partiendo de* $u^0 = {}^t(1,1)$.

Solución: hacemos iteraciones del esquema

$$Au^n = \frac{u^{n-1}}{\|u^{n-1}\|},$$

$$\begin{pmatrix} -2 & 1 \\ 0 & 3 \end{pmatrix} u^1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \quad \to u^1 = \begin{pmatrix} -\frac{1}{6}\sqrt{2} \\ \frac{1}{6}\sqrt{2} \end{pmatrix}, \left\|u^1\right\| = \frac{1}{3} = 0.33333,$$

$$\begin{pmatrix} -2 & 1 \\ 0 & 3 \end{pmatrix} u^2 = \begin{pmatrix} -\frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} \end{pmatrix} \quad \to u^2 = \begin{pmatrix} \frac{1}{3}\sqrt{2} \\ \frac{1}{6}\sqrt{2} \end{pmatrix}, \left\|u^2\right\| = \frac{1}{6}\sqrt{10} = 0.52705,$$

$\left\|u^2\right\|$ es el autovalor máximo de A^{-1}, con lo que el autovalor mínimo de A es $\lambda_{min} = \frac{-1}{\|u^2\|} = -\frac{6}{10}\sqrt{10} = -1.8974$, con signo negativo ya que $sign\left(\left\langle u^2, u^1\right\rangle\right) = -1$.

Problema 85 *Calcular el autovalor y autovector más cercano a* 2 *de la matriz*

$$\begin{pmatrix} 0 & -1 & 0 \\ 0 & 3 & -1 \\ 0 & 0 & -1 \end{pmatrix},$$

para ello calcular dos iteraciones del método de la potencia inversa partiendo de $u^0 = {}^t(1,1,1)$.

Solución:

$$A' = A - 2Id = \begin{pmatrix} -2 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & -3 \end{pmatrix}.$$

Vamos a utilizar la norma infinito con el fin de simplificar los cálculos. Hacemos iteraciones del esquema $A'u^n = \frac{u^{n-1}}{\|u^{n-1}\|}$

$$\begin{pmatrix} -2 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & -3 \end{pmatrix} u^1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \to u^1 = \begin{pmatrix} -\frac{5}{6} \\ \frac{2}{3} \\ -\frac{1}{3} \end{pmatrix}, \left\|u^1\right\| = \frac{5}{6} = 0.83333,$$

$$\begin{pmatrix} -2 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & -3 \end{pmatrix} u^2 = \frac{6}{5}\begin{pmatrix} -\frac{5}{6} \\ \frac{2}{3} \\ -\frac{1}{3} \end{pmatrix} \quad \to u^2 = \begin{pmatrix} \frac{1}{30} \\ \frac{14}{15} \\ \frac{2}{15} \end{pmatrix}, \left\|u^2\right\| = \frac{14}{15}.$$

El autovalor máximo de $(A - 2Id)^{-1}$ es $\lambda_{\text{máx}} = \frac{14}{15}$ con signo positivo $(sign\left(\left\langle u^2, u^1\right\rangle\right) = 1)$. Por tanto el autovalor más cercano a 2 de la matriz original es

$$\lambda = 2 + \frac{1}{\frac{14}{15}} = \frac{43}{14} = 3.0714,$$

y su autovector normalizado es $u^2 / \left\|u^2\right\|$.

Problema 86 *Calcular 3 iteraciones del método de Jacobi para resolver el sistema*

$$\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \\ 1 \end{pmatrix},$$

partiendo de $u^0 = {}^t(0, 0, 0)$.

Solución: Despejamos la diagonal para plantear el método iterativo :

$$x_n = y_{n-1} + 1,$$
$$y_n = \frac{x_{n-1} + 3}{2},$$
$$z_n = \frac{y_{n-1} + 1}{3},$$

haciendo iteraciones obtenemos

$$u^1 = \begin{pmatrix} -1 \\ \frac{3}{2} \\ \frac{1}{3} \end{pmatrix}, \qquad u^2 = \begin{pmatrix} \frac{1}{2} \\ 1 \\ \frac{5}{6} \end{pmatrix}, \qquad u^3 = \begin{pmatrix} 0 \\ \frac{7}{4} \\ \frac{2}{3} \end{pmatrix}.$$

Problema 87 *Calcular* 3 *iteraciones del método de Gauss-Seidel para resolver el sistema*

$$\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \\ 1 \end{pmatrix},$$

partiendo de $u^0 = {}^t(0,0,0)$.

Solución: Despejamos la diagonal para plantear el método iterativo, teniendo en cuenta además que vamos actualizando los valores según los calculamos:

$$\begin{aligned} x_n &= y_{n-1} + 1, \\ y_n &= \frac{x_n + 3}{2}, \\ z_n &= \frac{y_n + 1}{3}, \end{aligned}$$

haciendo iteraciones partiendo de ${}^t(0,0,0)$:

$$\begin{array}{lll} x_1 = -1, & x_2 = 0, & x_3 = -1 + \frac{3}{2} = \frac{1}{2}, \\ y_1 = \frac{3-1}{2} = 1, & \rightarrow \quad y_2 = \frac{3}{2}, & \rightarrow \quad y_3 = \frac{3+\frac{1}{2}}{2} = \frac{7}{4}, \\ z_1 = \frac{1+1}{3} = \frac{2}{3}, & z_3 = \frac{5}{6}, & z_3 = \frac{1+\frac{7}{4}}{3} = \frac{11}{12}. \end{array}$$

Problema 88 *Calcular* 3 *iteraciones del método de relajación para resolver el sistema*

$$\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \\ 1 \end{pmatrix},$$

partiendo de $u^0 = {}^t(0,0,0)$. *Calcular previamente el parámetro de relajación óptimo.*

Solución: el sistema puede escribirse como:

$$\begin{aligned} x - y &= -1, \\ -x + 2y &= 3, \\ -y + 3z &= 1. \end{aligned}$$

Cálculo del w_{opt}: al ser A tridiagonal, el w_{opt} se puede calcular como

$$w_{opt} = \frac{2}{1 + \sqrt{1 - \rho\left(M_J\right)^2}}.$$

M_J es la matriz del método de Jacobi que se obtiene despejando la diagonal en el sistema

$$\begin{array}{l} x = y - 1 \\ y = \frac{x}{2} + \frac{3}{2} \\ z = \frac{y}{3} + \frac{1}{3} \end{array} = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \end{pmatrix}}_{M_J} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} -1 \\ \frac{3}{2} \\ \frac{1}{3} \end{pmatrix}.$$

Los autovalores de M_J son $0, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}$, luego $\rho(M_J)^2 = \frac{1}{2}$ y por tanto

$$w_{opt} = \frac{2}{1 + \sqrt{1 - \rho(M_J)^2}} = \frac{2}{1 + \sqrt{1 - \frac{1}{2}}} = 1.1716.$$

Hacemos ahora iteraciones del sistema:

$$x_n = w_{opt}(y_{n-1} - 1) + (1 - w_{opt}) x_{n-1},$$

$$y_n = w_{opt} \tfrac{3 + x_n}{2} + (1 - w_{opt}) y_{n-1},$$

$$z_n = w_{opt} \tfrac{1 + y_n}{3} + (1 - w_{opt}) z_{n-1},$$

$$u^1 = \begin{pmatrix} -w_{opt} \\ w_{opt} \frac{3 - w_{opt}}{2} \\ w_{opt} \frac{1 + 1.0711}{3} \end{pmatrix} = \begin{pmatrix} -1.1716 \\ 1.0711 \\ 0.80883 \end{pmatrix},$$

$$u^2 = \begin{pmatrix} w_{opt}(1.0711 - 1) - (1 - w_{opt}) 1.1716 \\ w_{opt} \frac{3 + .28435}{2} + (1 - w_{opt}) 1.0711 \\ w_{opt} \frac{1 + 1.7402}{3} + (1 - w_{opt}) 0.80883 \end{pmatrix} = \begin{pmatrix} 0.28435 \\ 1.7402 \\ 0.93134 \end{pmatrix},$$

$$u^3 = \begin{pmatrix} w_{opt}(1.7402 - 1) + (1 - w_{opt}) 0.28435 \\ w_{opt} \frac{3 + 0.81842}{2} + (1 - w_{opt}) 1.7402 \\ w_{opt} \frac{1 + 1.9382}{3} + (1 - w_{opt}) 0.93134 \end{pmatrix} = \begin{pmatrix} 0.81842 \\ 1.9382 \\ 0.98765 \end{pmatrix},$$

Problema 89 *Demostrar que si una matriz A verifica que por filas o columnas su suma es siempre igual a 0, entonces el determinante de A es cero.*

Solución: Vamos a demostrar que si la suma por filas de A es igual a cero, entonces $|A| = 0$. Efectivamente si la suma por filas es cero entonces

$$A \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = 0 \cdot \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

esto significa que la matriz A posee un autovalor igual a cero ($\lambda = 0$). Como el determinante de una matriz es igual al producto de sus autovalores obtenemos:

$$|A| = \prod_{i=1}^{n} \lambda_i = \lambda_1 \cdot \ldots\ldots \cdot \lambda_n = 0,$$

por otro lado si la suma por columnas es 0, basta tener en cuenta que $|A| = \left|{}^tA\right|$ y aplicar el argumento anterior a ${}^tA$.

Problema 90 *Demostrar que dado un sistema iterativo*

$$u^n = Mu^{n-1} + c,$$

aunque el radio espectral de M sea mayor que 1, si u^1 y c son combinaciones lineales de autovectores de M correspondientes a autovalores de módulo menor que 1, entonces u^n converge.

Solución: Sean x_i los autovectores de M correspondientes a autovalores menores que 1:

$u^1 = \sum_{i=1}^n a_i x_i,$

$c = \sum_{i=1}^n c_i x_i.$

Realizando iteraciones obtenemos las siguientes expresiones:

$u^2 = Mu^1 + c,$

$u^3 = Mu^2 + c = M\left(Mu^1 + c\right) + c = M^2u^1 + Mc + c,$

$\vdots$

$u^n = M^{n-1}u^1 + M^{n-2}c + \ldots Mc + c = M^{n-1}u^1 + \left(M^{n-2} + \ldots M + 1\right)c.$

Tomando el primer sumando:

$M^{n-1}u^1 = M^{n-1}\sum_{i=1}^n a_i x_i = M^{n-2}\sum_{i=1}^n a_i M x_i = M^{n-2}\sum_{i=1}^n a_i \lambda_i x_i =$
$= \ldots \sum_{i=1}^n a_i \lambda_i^{n-1} x_i.$

Como u^1 depende linealmente de los x_i (autovectores) cuyos autovalores λ_i son menores que uno, entonces λ_i^{n-1} tiende a 0 cuando n tiende a infinito, luego este término converge.

Para el segundo sumando:

$$\begin{aligned}
&\left(M^{n-2} + \ldots M + 1\right)c = \left(M^{n-2} + \ldots M + 1\right)\sum_{i=1}^n c_i x_i = \\
&= M^{n-2}\sum_{i=1}^n c_i x_i + \ldots M\sum_{i=1}^n c_i x_i + \sum_{i=1}^n c_i x_i \\
&= \sum_{i=1}^n c_i \lambda_i^{n-2} x_i + \ldots \sum_{i=1}^n c_i \lambda_i x_i + \sum_{i=1}^n c_i x_i = \\
&= \sum_{i=1}^n c_i x_i \underbrace{\left(\lambda_i^{n-2} + \ldots + \lambda_i + 1\right)}_{\text{Serie geométrica convergente}} \leq \sum_{i=1}^n c_i x_i \tfrac{1}{1-\lambda_i},
\end{aligned}$$

con lo que este término también converge. Hay que tener en cuenta que este resultado es teórico trabajando sin errores de redondeo en las operaciones. En el caso de trabajar con una aritmética de precisión finita y que existan autovalores de módulo

mayor que 1, es posible que los errores que se van acumulando provoquen que u^n ya no sea combinación lineal de autovectores de M correspondientes a autovalores de módulo menor que 1 y por tanto el algoritmo numérico puede diverger.

Problema 91 *Calcular* 2 *iteraciones del método de Newton-Raphson no-lineal para aproximar una raíz del sistema de ecuaciones*

$$\begin{aligned} x^2+y^2-1&=0,\\ y-x&=0, \end{aligned}$$

partiendo de $u^0={}^t(1,1)$.

Solución: la matriz gradiente del sistema no-lineal viene dada por

$$\nabla f(x,y)=\begin{pmatrix} 2x & 2y\\ -1 & 1\end{pmatrix},$$

tenemos que hacer iteraciones del esquema

$$\begin{cases} \nabla f(u^n)z=-f(u^n),\\ u^{n+1}=u^n+z. \end{cases}$$

Iteraciones:

1. $\begin{pmatrix} 2 & 2\\ -1 & 1\end{pmatrix}\begin{pmatrix} z_1\\ z_2\end{pmatrix}=-\begin{pmatrix} 1\\ 0\end{pmatrix}\to\begin{pmatrix} z_1\\ z_2\end{pmatrix}=\begin{pmatrix} -\frac{1}{4}\\ -\frac{1}{4}\end{pmatrix}\to u^1=\begin{pmatrix} 1\\ 1\end{pmatrix}+\begin{pmatrix} -\frac{1}{4}\\ -\frac{1}{4}\end{pmatrix}\begin{pmatrix} \frac{3}{4}\\ \frac{3}{4}\end{pmatrix},$

2. $\begin{pmatrix} \frac{3}{2} & \frac{3}{2}\\ -1 & 1\end{pmatrix}\begin{pmatrix} z_1\\ z_2\end{pmatrix}=-\begin{pmatrix} \frac{1}{8}\\ 0\end{pmatrix}\to\begin{pmatrix} z_1\\ z_2\end{pmatrix}=\begin{pmatrix} -\frac{1}{24}\\ -\frac{1}{24}\end{pmatrix}\to u^2=\begin{pmatrix} \frac{3}{4}\\ \frac{3}{4}\end{pmatrix}+\begin{pmatrix} -\frac{1}{24}\\ -\frac{1}{24}\end{pmatrix}=\begin{pmatrix} \frac{17}{24}\\ \frac{17}{24}\end{pmatrix},$

Problema 92 *Plantear el algoritmo necesario para calcular, utilizando el método de Newton-Raphson, las raíces complejas de un polinomio de grado* 3.

Solución:

$$P(z)=az^3+bz^2+cz+d=0,$$

un polinomio de grado 3 posee al menos una raíz real. Las otras dos raíces pueden ser también reales o complejas conjugadas. Sea z un número complejo: $z=x+yi$, sustituyendo en la ecuación anterior,

$$P(x+yi)=a(x+yi)^3+b(x+yi)^2+c(x+yi)+d,$$

quitamos paréntesis en esta expresión y obtenemos

$$P(x+yi)=ax^3+3iax^2y-3axy^2-iay^3+bx^2+2ibxy-by^2+cx+icy+d=0.$$

Separamos la parte real de la parte imaginaria:

$$f = \left\{ \begin{array}{l} ax^3 - 3axy^2 + bx^2 - by^2 + cx + d = 0, \\ 3ax^2y - ay^3 + 2bxy + cy = 0, \end{array} \right.$$

la matriz gradiente es

$$\nabla f = \left(\begin{array}{cc} 3ax^2 - 3ay^2 + 2bx + c & -6axy - 2by \\ 6axy + 2by & 3ax^2 - 3ay^2 + 2bx + c \end{array} \right),$$

Por tanto el algoritmo consiste en hacer iteraciones del esquema:

$$\left\{ \begin{array}{l} \nabla f\left(u^n\right) \cdot z = -f\left(u^n\right), \\ u^{n+1} = u^n + z, \end{array} \right.$$

para este caso particular de función f.

Problema 93 *Se considera el sistema no-lineal*

$$\begin{aligned} (x-1)y &= 0, \\ (y-2)x &= 0. \end{aligned}$$

A partir de $u^0 = {}^t(1,1)$*, calcular* u^1 *y* u^2 *utilizando el método de Newton-Raphson para aproximar un cero del sistema no-lineal.*

Solución:

$$\nabla f(x,y) = \left(\begin{array}{cc} y & x-1 \\ y-2 & x \end{array} \right),$$

$$\nabla f(1,1) = \left(\begin{array}{cc} 1 & 0 \\ -1 & 1 \end{array} \right) \rightarrow u^1 = \left(\begin{array}{cc} 1 & 0 \\ -1 & 1 \end{array} \right)^{-1} \left(\begin{array}{c} 0 \\ 1 \end{array} \right) + \left(\begin{array}{c} 1 \\ 1 \end{array} \right) = \left(\begin{array}{c} 1 \\ 2 \end{array} \right),$$

$$\nabla f(1,2) = \left(\begin{array}{cc} 2 & 0 \\ 0 & 1 \end{array} \right) \rightarrow u^2 = \left(\begin{array}{cc} 2 & 0 \\ 0 & 1 \end{array} \right)^{-1} \left(\begin{array}{c} 0 \\ 0 \end{array} \right) + \left(\begin{array}{c} 1 \\ 2 \end{array} \right) = \left(\begin{array}{c} 1 \\ 2 \end{array} \right),$$

Problema 94 *Calcular* 1 *iteración del método de Newton-Raphson no-lineal para aproximar una raíz del sistema de ecuaciones*

$$\begin{aligned} e^{xyz} - 1 &= 0, \\ y^2 - z^3 - 2 &= 0, \\ (z-1)x^4 - 3 &= 0, \end{aligned}$$

partiendo de $u^0 = {}^t(1,1,1)$.

Solución:

$$\nabla f(x,y,z) = \begin{pmatrix} yze^{xyz} & xze^{xyz} & xye^{xyz} \\ 0 & 2y & -3z^2 \\ 4(z-1)x^3 & 0 & x^4 \end{pmatrix},$$

hay que hacer iteraciones del esquema

$$\begin{cases} \nabla f(x,y,z)\, z = -f(x,y,z), \\ u^{n+1} = u^n + z. \end{cases}$$

1^a iteración partiendo de $u^0 = {}^t(1,1,1)$:

$$\begin{pmatrix} e & e & e \\ 0 & 2 & -3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = - \begin{pmatrix} e-1 \\ -2 \\ -3 \end{pmatrix} \quad \rightarrow \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} -\frac{1}{e}(e-1) - \frac{17}{2} \\ \frac{11}{2} \\ 3 \end{pmatrix}$$

$$u^1 = u^0 + z = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} -\frac{1}{e}(e-1) - \frac{17}{2} \\ \frac{11}{2} \\ 3 \end{pmatrix} = \begin{pmatrix} -\frac{15}{2} - \frac{1}{e}(e-1) \\ \frac{13}{2} \\ 4 \end{pmatrix}.$$

Problema 95 *Se considera la función $F(x,y) = x^4 + (x-1)^2 + (y-1)^2$. Aplicar una iteración del método de Newton-Raphson a $\nabla F(x,y)$ para optimizar $F(x,y)$ partiendo de $(x_0,y_0) = (1,1)$. Estudiar si disminuye el valor de la función en el nuevo punto calculado.*

Solución: Se calcula $f(x,y) = \nabla F(x,y)$ y $\nabla f(x,y)$ obteniendo

$$f(x,y) = \nabla F(x,y) = \begin{pmatrix} 4x^3 + 2x - 2 \\ 2y - 2 \end{pmatrix} \qquad \nabla f(x,y) = \begin{pmatrix} 12x^2 + 2 & 0 \\ 0 & 2 \end{pmatrix}$$

el aplicar el método de Newton-Raphson a partir de $(x,y) = (1,1)$ nos lleva al sistema de ecuaciones

$$\begin{pmatrix} 12+2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 - 1 \\ y_1 - 1 \end{pmatrix} = - \begin{pmatrix} 4+2-2 \\ 2-2 \end{pmatrix}$$

cuya solución es $(x_1, y_1) = (\frac{5}{7}, 1)$. El valor de la función en el punto inicial es $F(1,1) = 1$. El valor de la función en el nuevo punto calculado es $F(\frac{5}{7}, 1) = \frac{821}{2401}$ que es menor que 1. Por tanto la función disminuye en el nuevo punto.

Problema 96 *Se considera la función $F(x,y) = x^4 + (x-1)^2 + (y-1)^2$. Aplicar una iteración completa del método de gradiente descendente para optimizar $F(x,y)$ partiendo de $(x_0,y_0) = (1,1)$, utilizando como valor inicial $\lambda = 1$.*

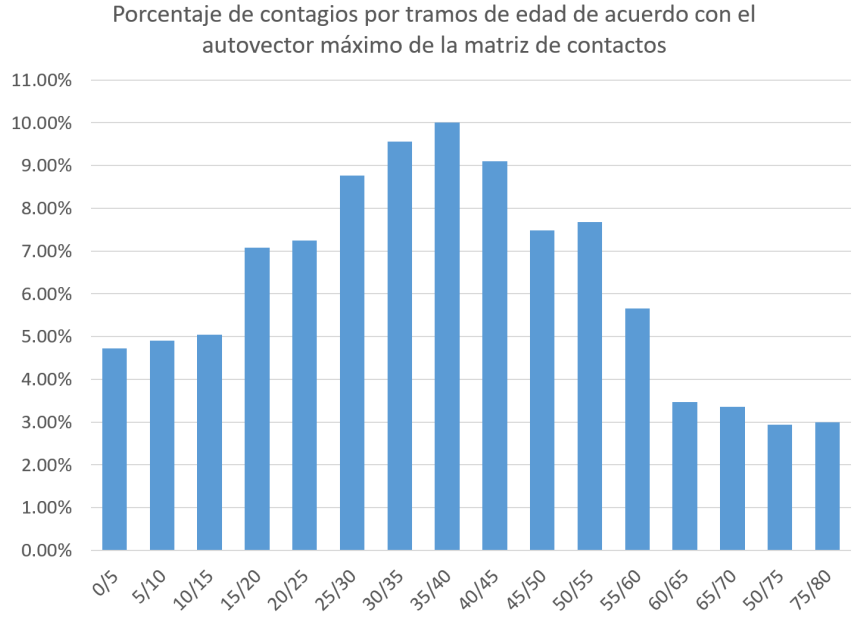
Solución: Se calcula el gradiente de la función $\nabla F(x,y) = {}^t\big(4x^3 + 2x - 2, 2y - 2\big)$. Al evaluarlo en $(1,1)$ nos da $\nabla F(1,1) =^t (4,0)$ por tanto el nuevo candidato para el mínimo será

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \lambda \begin{pmatrix} 4 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 - 4\lambda \\ 1 \end{pmatrix}$$

La función en el punto inicial es $F(1,1) = 1$. Al hacer $\lambda = 1$ obtenemos $(x_1, y_1) = (-3, 1)$ y $F(-3,1) = 97 > F(1,1)$, por tanto este candidato lo desechamos y tomamos ahora $\lambda = 1/10$. En ese caso $(x_1, y_1) = (\frac{3}{5}, 1)$ y $F(\frac{3}{5}, 1) = \frac{181}{625} < F(1,1)$ y por tanto este candidato es válido y la solución después de completar la primera iteración es $(x_1, y_1) = (\frac{3}{5}, 1)$.

6.8. Aplicación en Epidemiología

En este apartado vamos a ver diferentes aplicaciones del Análisis Matricial y la Optimización en Epidemiología, empezaremos por el análisis de la distribución por tramos de edad de los contagios usando la matriz de contactos, A. Como se vió en el tema 4, para pasar de una generación de contagios u^n a la siguiente generación u^{n+1} se tiene en cuenta que u^{n+1} va a ser proporcional a Au^n y que la suma de los elementos de u^{n+1} debe ser R veces la suma de los elementos de u^n (donde R es la tasa de reproducción). Ahora bien la operación de ir haciendo Au^n de forma iterativa es justamente la operación que se hace en el método de la potencia visto en este tema. Como se vió en este método, el vector $u^n/||u^n||$ converge hacia el autovector de mayor valor absoluto de la matriz A. Por tanto, tomando la norma 1 de vectores (es decir la suma de las componentes del vector) este autovector representa la proporción (o porcentaje), de como se van distribuyendo los contagios por tramos de edad. En la gráfica siguiente se ilustra esta distribución



además, se puede comprobar que el autovalor máximo de la matriz A es simple, es decir, tiene asociado un único autovector linealmente independiente (aunque en este curso no se han dado las herramientas para demostrar esto, dado que no hemos visto ningún método para calcular todos los autovectores de matrices no simétricas). En cualquier caso, si damos por cierto que el autovalor máximo de A es simple y tiene asociado un único autovector entonces el valor final de las iteraciones no depende del vector de salida u_0, dicho de otro modo, independientemente de cual sea la franja de edad del paciente cero, la distribución de la proporción de contagios por franjas de edad a la que se va convergiendo es la misma.

La segunda aplicación que vamos a estudiar es el estudio del crecimiento inicial del nº de contagios diarios registrados a través de los test. El estudio de este crecimiento inicial cuando el virus circula libremente al principio de la epidemia es de gran importancia, pues entre otras cosas determina la tasa de reproducción inicial R, o el porcentaje de personas inmunizadas que hace falta para alcanzar la inmunidad de grupo, etc.. La hipótesis de partida que se hace habitualmente es que el crecimiento inicial sigue un modelo exponencial del tipo $y = ae^{cx} + b$, donde a, b y c son parámetros del modelo. El parámetro más importante desde el punto de vista epidemiológico es c que representa la tasa de crecimiento exponencial. Para calcular a, b y c se plantea el siguiente problema de optimización:

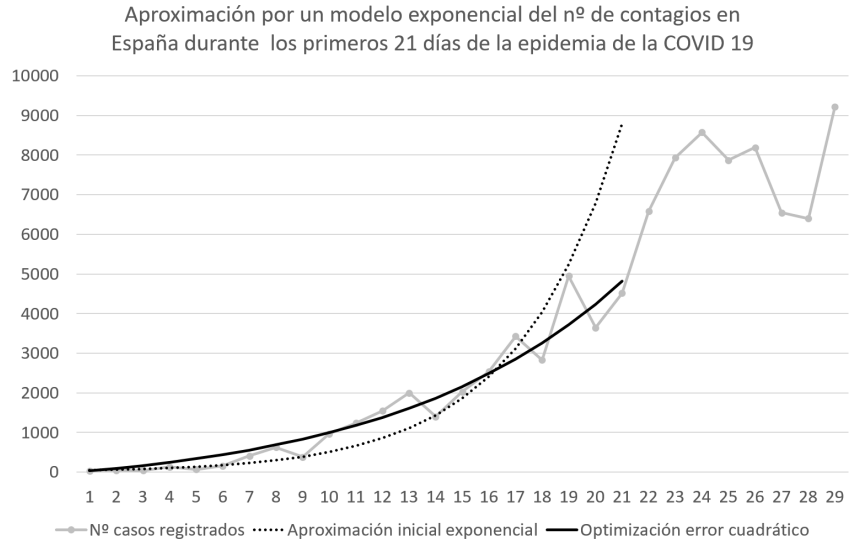
$$F(a,b,c) = \sum_{k=0}^{N-1} (ae^{ck} + b - y_k)^2 \tag{6.137}$$

donde $F(a,b,c)$ es la función objetivo y y_k es el número de contagios registrados al principio de la epidemia. Nosotros tomaremos los 21 primeros días y consideramos que el principio de la epidemia se produce cuando se registra más de 1 contagio diario por millón de habitantes. Para aplicar los métodos de optimización vistos en este curso, necesitamos aportar una aproximación inicial de los parámetros a, b y c. Para ello, tomaremos inicialmente $b = 0$ y tenemos en cuenta que si $y_k \approx ae^{ck}$, entonces tomando logaritmos tendremos que $ln(y_k) \approx ln(a) + ck$. Por tanto si aproximamos $(x, ln(y_k))$ usando un modelo de regresión lineal a través de la recta $y = mx + n$ tendremos que $a = e^n$ y $c = m$, y esto nos daría una aproximación inicial de los parámetros. Hay que tener en cuenta que al optimizar a, b y c a través del modelo de optimización anterior podríamos tener resultados físicamente imposibles como que al evaluar la aproximación $ae^{ck} + b$ nos diese un número negativo para algún $k \geq 0$. Una forma sencilla de evitar que esto ocurra es penalizar en la función objetivo que la aproximación de y_0 salga un número menor que y_0. Es decir, queremos que la aproximación arranca en y_0 o en un número superior. Para ello, lo que podemos hacer es usar como función objetivo:

$$F(a,b,c) = \sum_{k=0}^{N-1} (ae^{ck} + b - y_k)^2 + \lambda(min\{a + b - y_0, 0\})^2 \tag{6.138}$$

donde $\lambda > 0$ representa la "fuerza" con la que queremos imponer la nueva condición. Cuanto más grande es λ más penalizamos que no se cumpla la condición de que

$ae^{c\cdot 0} + b = a + b \geq y_0$. En la siguiente gráfica se ilustran los resultados obtenidos para el caso de España. Se presenta el nº de casos registrados comunicados por el gobierno durante los primeros 30 días de la epidemia, para la optimización se usan los primeros 21 días y se presenta la aproximación inicial del modelo a partir de la regresión lineal y el resultado de la optimización usando la función objetivo (6.138) con $\lambda = 10^4$ y el método de Newton-Raphson atenuado.



Como puede observarse, el resultado obtenido a partir de la optimización se ajusta mucho mejor a los datos que la aproximación inicial. El análisis del crecimiento inicial de los contagios es un problema complejo, no por el modelo matemático utilizado para aproximar los datos, sino por la mala calidad de los datos usados. Algunos de los motivos de esa mala calidad son: (1) al principio de la epidemia, la capacidad para hacer test de los países es bastante limitada, se hacen pocos test y por tanto muchos contagiados no son registrados, (ii) al principio, los pacientes que se registran son principalmente los que presentan síntomas, por tanto de los asintomáticos se detectan muy pocos, (iii) al principio se tardaba mucho entre que un paciente presentaba síntomas, se le hacía el test y finalmente se registraba su caso, además ese retraso resultaba muy variable, por tanto el dato de los contagiados registrados un día registra casos de contagios de mucho días diferentes esto además se amplifica teniendo en cuenta que el tiempo que tarda una persona en presentar síntomas es bastante variable. Por ello, en los estudios científicos realizados no hay unanimidad en los resultados obtenidos pues debido a esta mala calidad de los datos, los resultados pueden depender bastante del país en concreto que se use para el estudio.

Lo siguiente que vamos a estudiar es lo que se denomina el "serial interval", de gran importancia en Epidemiología, que corresponde a la distribución estadística del tiempo que pasa entre que un caso primario presenta síntomas y un caso secundario (persona contagiada por el caso primario) presenta síntomas. Para ello, lo primero que hay que hacer es tomar una muestra de pares de casos donde se pueda asociar el caso

primario al caso secundario y se haya registrado el momento en que cada uno de ellos presentó síntomas. Esto se hizo en [Ma] sobre una muestra de 689 casos. En las tablas siguientes se muestran los resultados obtenidos: en la fila superior se pone el número de días que han pasado entre que el caso primario y el secundario presentan síntomas (nótese que un valor negativo indica que el caso secundario presentó síntomas antes que que el caso primario) y en la fila inferior el número de casos registrados para ese número de días. Por ejemplo, hubo 2 casos donde el caso secundario presentó síntomas 5 días antes del caso primario y hubo 63 casos donde el caso secundario presentó síntomas 3 días después que el caso primario.

días entre casos	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9
nº de casos	2	5	9	4	8	32	38	50	63	62	44	58	50	40	39

días entre casos	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
nº de casos	33	31	18	17	18	16	17	15	7	4	3	3	1	1	1

Como cualquier muestra es imperfecta para aproximar el "serial interval" ajustaremos la muestra a una distribución de probabilidad conocida. En este caso vamos a usar la distribución log-normal, propuesta en [Ma] y que es una distribución que se usa habitualmente en Epidemiología al igual que la distribución Gamma vista anteriormente. La forma general de la distribución log-normal es la siguiente:

$$f(x) = \begin{cases} 0 & \text{si} \quad x \leq a \\ d\,\dfrac{e^{-\frac{(\log(x-a)-\mu)^2}{2\sigma^2}}}{(x-a)\sigma\sqrt{2\pi}} & \text{si} \quad x > a \end{cases} \tag{6.139}$$

la distribución tiene 4 parámetros: μ, σ, a que representa un desplazamiento y d que es un factor de escala para ajustar la distribución al tamaño de la muestra de tal forma que se cumple:

$$\int_a^\infty f(x)dx = d \tag{6.140}$$

dada la muestra que tenemos para el "serial interval", lo primero que tenemos que hacer es buscar una estimación inicial de los parámetros de la log-normal. Para ello vamos a tomar inicialmente $a = -5$ que es el primer valor para el cual hay casos y $d = 689$, que es el tamaño de la muestra. Para estimar μ y σ, calculamos primero la media, m, y varianza V muestrales y ajustamos a continuación μ y σ para que la distribución log-normal tenga la misma media y varianza. Teniendo en cuenta el valor de la media y varianza de la log-normal, ello nos da las ecuaciones

$$m = a + e^{\mu + \frac{\sigma^2}{2}} \tag{6.141}$$

$$V = \left(e^{\mu+\frac{\sigma^2}{2}}\right)^2 \left(e^{\sigma^2} - 1\right) \tag{6.142}$$

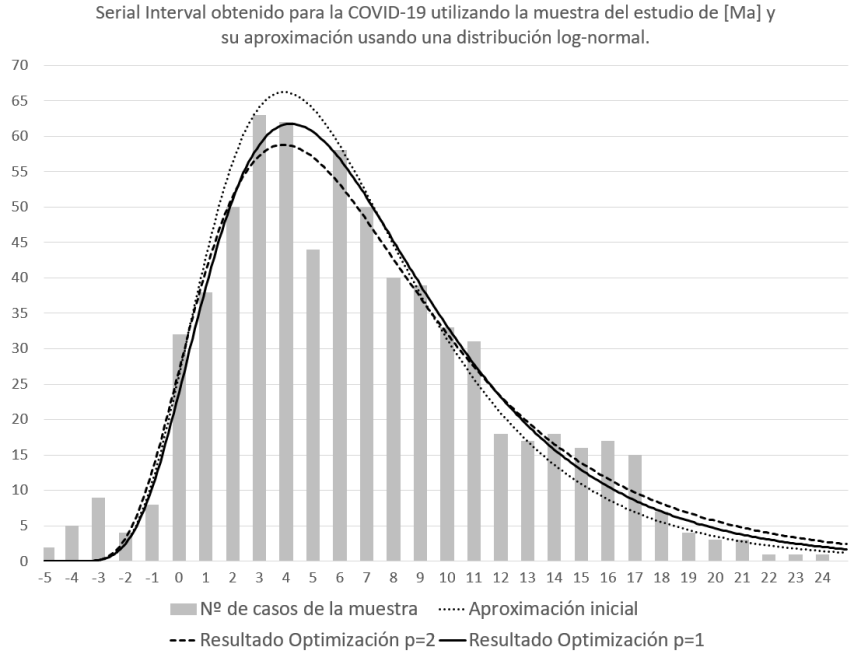
de estas ecuaciones podemos despejar μ y σ obteniendo:

$$\sigma = \sqrt{\ln\left(\frac{V}{m^2}+1\right)} \qquad (6.143)$$
$$\mu = \ln(m) - \frac{\sigma^2}{2}$$

para mejorar la estimación inicial de los parámetros, formulamos un problema de optimización donde la función objetivo es

$$F(\mu,\sigma,a,b) = \sum_k |y_k - f(x_k)|^p \qquad (6.144)$$

donde (x_k, y_k) son los valores que determinan la muestra, es decir $(x_0, y_0) = (-5, 2)$, $(x_1, y_1) = (-4, 5)$ y así sucesivamente. $p > 0$ es un parámetro de la función objetivo. Los valores más usuales para p son $p = 2$ que determina una función objetivo cuadrática o $p = 1$ donde se minimiza la diferencia en valor absoluto entre el valor de la log-normal y la muestra. En la gráfica siguiente se ilustra el resultado de la aproximación de dicha muestra por la distribución log-normal usando la aproximación inicial y la optimización con $p = 2$ y $p = 1$ en la función objetivo (se usó el método de Newton-Raphson atenuado). Se aprecia como en la aproximación inicial el pico de la log-normal sube demasiado respecto a la muestra. El resultado de la optimización mejora la aproximación y tanto para $p = 2$ como para $p = 1$ los resultados son razonables.



Un problema interesante que se plantea es que distribución de probabilidad es la mejor que se ajusta a la muestra. Por ejemplo ¿hubiese sido mejor aproximar el "serial

interval" por una distribución Gamma?. Usando la función objetivo de la optimización esto es una cuestión fácil de resolver pues podemos optimizar usando ámbas distribuciones de probabilidad y quedarnos con aquella para la que obtengamos un menor valor de la función objetivo. Nótese que a la distribución Gamma que vimos en el tema 2 habría que añadirle un parámetro a de desplazamiento para poder aproximar una muestra como la del "serial interval", es decir, la función de densidad de la distribución Gamma sería :

$$f(x) = \begin{cases} 0 & \text{si} \quad x \leq a \\ d(x-a)^{\alpha-1} e^{-\beta(x-a)} & \text{si} \quad x > a \end{cases} \tag{6.145}$$

Capítulo 7

INTERPOLACIÓN DE FUNCIONES II

En este tema veremos algunos aspectos más avanzados de la interpolación de funciones. Concretamente veremos la interpolación de Hermite que es una generalización de la interpolación de Lagrange donde además de fijar el valor de la función en los puntos de interpolación también se ajusta el valor de sus derivadas, la interpolación usando la función seno cardinal y la interpolación usando polinomios trigonométricos de gran importancia en las aplicaciones de procesado de la señal, en particular representan la base para el almacenamiento y transmisión del sonido digital.

7.1. Interpolación de Hermite.

En ocasiones, resulta de interés interpolar no sólo el valor de la función en ciertos puntos $\{x_i\}_{i=0,..,N}$, sino también el valor de sus derivadas. Un ejemplo clásico de ello es el desarrollo de Taylor de una función en un punto a. En este caso, aproximamos $f(x)$ por un polinomio de grado N, $P_N(x)$ tal que $f(x)$ y $P_N(x)$ poseen las mismas derivadas en el punto a desde el orden 0 hasta el orden N.

$$P_N(x) = f(a) + \frac{f'(a)}{1!}(x-a) + .. + \frac{f^{N)}(a)}{N!}(x-a)^N. \tag{7.1}$$

El error de interpolación viene dado por la fórmula

$$f(x) - P_N(x) = \frac{f^{N+1)}(\xi)}{(N+1)!}(x-a)^{N+1}, \tag{7.2}$$

donde ξ es un valor intermedio entre x y a. En el caso general, donde buscamos un polinomio $P(x)$ tal que él y todas sus derivadas hasta un cierto orden M coincidan con una función $f(x)$ en los puntos $\{x_i\}_{i=0,..,N}$, se utilizan los denominados polinomios base de Hermite $H_{i,j}(x)$, que son polinomios de grado menor o igual que

$(N+1)(M+1)-1$ dados por las siguientes condiciones:

$$\frac{\partial^l H_{i,j}}{\partial x^l}(x_k) = \begin{cases} 1 & \text{si} \quad l = j \ y \ k = i, \\ 0 & \text{si} \quad l \neq j \ ó \ k \neq i. \end{cases} \tag{7.3}$$

A partir de los polinomios base de Hermite, el polinomio interpolador se define como:

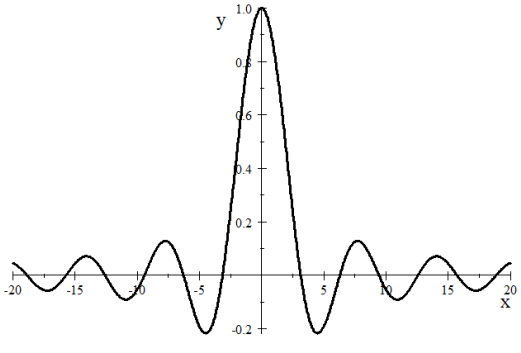
$$P(x) = \sum_{i=0}^{N} \sum_{j=0}^{M} \frac{\partial^j f}{\partial x^j}(x_i) H_{i,j}(x). \tag{7.4}$$

7.2. La interpolación a través de la función seno cardinal.

Una base de funciones interpolantes muy utilizada en el tratamiento de la señal es la base definida a partir de la función seno cardinal, definida por

$$\text{sinc}(x) = \frac{sen(x)}{x}, \tag{7.5}$$

cuya gráfica es



Esta función tiene la propiedad de que $\text{sinc}(0) = 1$, y para cualquier entero i distinto de 0, $\text{sinc}(\pi i) = 0$. Dada una función $f(x)$, su función interpolante en los puntos $x_i = a \cdot i$ para i números enteros variando entre dos valores $n_0 < n_1$ viene dada por la función

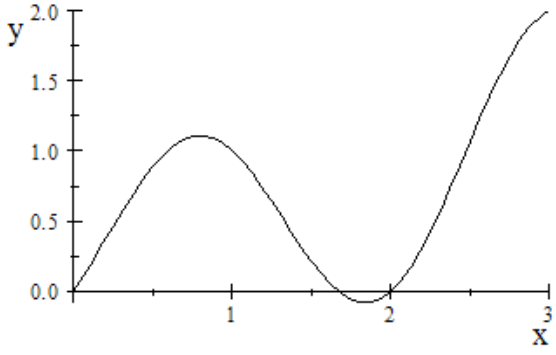
$$\widetilde{f}(x) = \sum_{i=n_0}^{n_1} f(x_i) \text{sinc}\left(\pi\left(\frac{x}{a} - i\right)\right), \tag{7.6}$$

a cada valor $f(x_i)$ se le llama muestra de la función en el punto x_i. Por ejemplo, al digitalizar una señal de sonido, lo que se hace es tomar muestras de la señal cada intervalo de tiempo a, de esta manera la señal digitalizada se puede almacenar en el ordenador, transmitir y posteriormente reproducir reconstruyéndola usando la fórmula anterior.

Ejemplo 19 *Consideremos la función $f(x)$, definida en los puntos $x = 0, 1, 2,$ y 3, tal que $f(0) = 0$, $f(1) = 1$, $f(2) = 0$, $f(3) = 2$. La interpolación de esta función utilizando la función seno cardinal viene dada por la función*

$$\widetilde{f}(x) = sinc\left(\pi\left(x-1\right)\right) + 2sinc\left(\pi\left(x-3\right)\right), \tag{7.7}$$

cuya gráfica es



7.3. La interpolación a través de polinomios trigonométricos.

La base de la transformada de Fourier discreta es la utilización de los polinomios trigonométricos dados por la expresión

$$P^k(x) = e^{ikx}. \tag{7.8}$$

Dada una función $f(x)$, definida en el intervalo $[-\pi, \pi]$, pretendemos aproximar $f(x)$ como

$$f(x) \approx \sum_{k=-N}^{N} c_k e^{ikx}, \tag{7.9}$$

donde c_k son coeficientes, en general, complejos. El siguiente resultado determina la forma de calcular dichos coeficientes c_k:

Teorema 36 *Los coeficientes c_k que minimizan el error cuadrático medio*

$$E(c_{-N}, ..., c_N) = \int_{-\pi}^{\pi} \left(f(x) - \sum_{k=-N}^{N} c_k e^{ikx} \right)^2 dx, \tag{7.10}$$

vienen dados por

$$c_k = \frac{\int_{-\pi}^{\pi} f(x) e^{-ikx} dx}{2\pi}. \tag{7.11}$$

Demostración: en primer lugar, observamos que, dada la forma cuadrática del funcional $E(c_{-N},...,c_N)$, éste debe poseer mínimos. Por otro lado, en un mínimo, las derivadas parciales de $E(c_{-N},...,c_N)$ con respecto a cualquier c_k son cero, y por tanto

$$\frac{\partial E}{\partial c_k}(c_{-N},...,c_N) = \int_{-\pi}^{\pi} \left(f(x) - \sum_{l=-N}^{N} c_l e^{ilx} \right) e^{ikx} dx = 0, \tag{7.12}$$

con lo que el resultado del teorema sale de forma inmediata, teniendo en cuenta que

$$\int_{-\pi}^{\pi} e^{ilx} e^{ikx} dx = \begin{cases} 2\pi & si \quad l = -k, \\ 0 & si \quad l \neq k. \end{cases} \tag{7.13}$$

Ejemplo 20 *Consideremos la función*

$$f(x) = \begin{cases} 1 & si \quad x \in [-\frac{\pi}{2}, \frac{\pi}{2}], \\ 0 & si \quad x \notin [-\frac{\pi}{2}, \frac{\pi}{2}]. \end{cases} \tag{7.14}$$

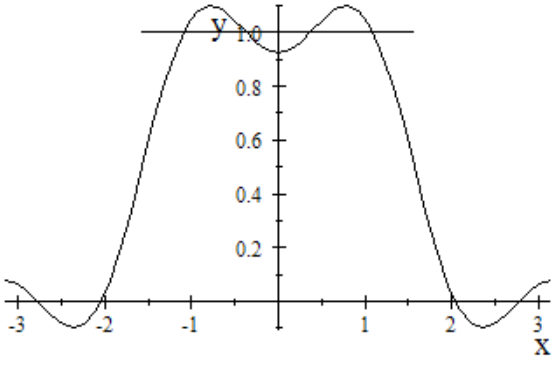
Vamos a calcular el polinomio trigonométrico interpolante para $N = 3$. Los valores de c_k son

$$c_0 = \frac{\int_{-\pi}^{\pi} f(x)dx}{2\pi} = \frac{1}{2}, \qquad c_1 = c_{-1} = \frac{\int_{-\pi}^{\pi} f(x)e^{ix}dx}{2\pi} = \frac{1}{\pi}, \tag{7.15}$$
$$c_2 = c_{-2} = \frac{\int_{-\pi}^{\pi} f(x)e^{2ix}dx}{2\pi} = 0, \qquad c_3 = c_{-3} = \frac{\int_{-\pi}^{\pi} f(x)e^{3ix}dx}{2\pi} = -\frac{1}{3\pi}.$$

Por tanto, el polinomio trigonométrico interpolador es

$$P_3(x) = \frac{1}{2} + \frac{2}{\pi}\cos(x) - \frac{2}{3\pi}\cos(3x). \tag{7.16}$$

La siguiente gráfica muestra la aproximación entre $f(x)$ y $P_3(x)$:

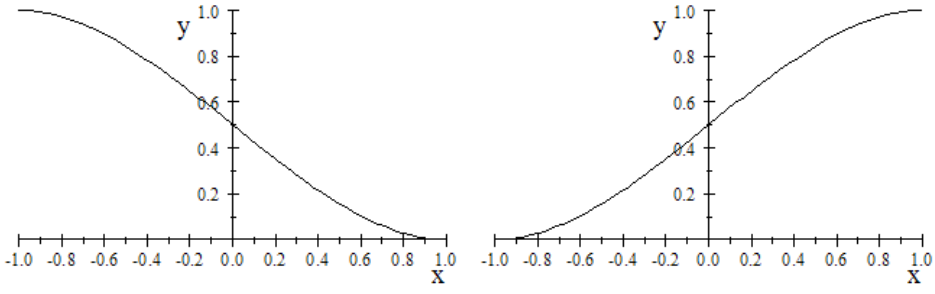


La interpolación por la función $\text{sinc}(x)$ y los polinomios trigonométricos son muy útiles en el ámbito de las aplicaciones de tratamiento de la señal como por ejemplo el sonido digital.

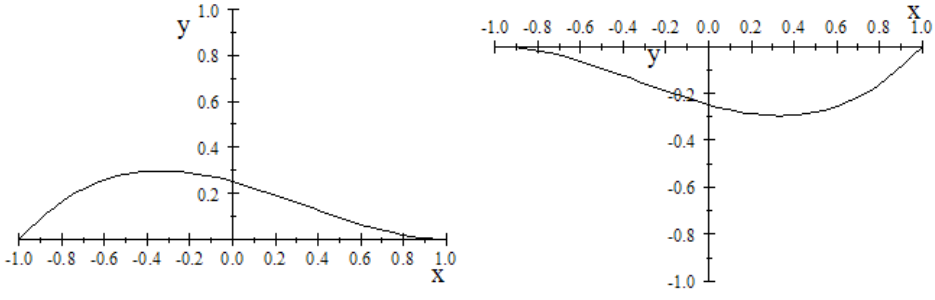
7.4. Problemas resueltos

Problema 97 *Calcular los polinomios base de Hermite que corresponden a tomar como puntos de interpolación* $x_0 = -1$, $x_1 = 1$, *y el orden de derivación* $M = 1$.

Solución: Los polinomios de Hermite que corresponden a esos puntos de interpolación vienen dados por las gráficas:



Polinomio de Hermite $H^0_{-1}(x)$ y $H^0_1(x)$



Polinomio de Hermite $H^1_{-1}(x)$ y $H^1_1(x)$.

De forma analítica se calculan de la forma siguiente:

1. H^0_{-1} se hace cero en 1 y sus derivadas, tanto en ese punto como en -1, valen cero. Por tanto este polinomio tiene una raíz doble en 1 y factorizando se puede expresar como:

$$H^0_{-1}(x) = (x-1)^2 (a(x+1)+b).$$

Por otro lado al imponer que $H^0_{-1}(-1) = 1$ y $\frac{dH^0_{-1}}{dx}(-1) = 0$ se obtiene

$$H^0_{-1}(-1) = (-1-1)^2 (a(-1+1)+b) = 4b = 1 \quad \rightarrow \quad b = \frac{1}{4},$$

$$\frac{dH^0_{-1}}{dx}(-1) = 2(-1-1)\left(a(-1+1)+\frac{1}{4}\right) + (-1-1)^2 a = 0 \quad \rightarrow \quad a = \frac{1}{4}.$$

Por tanto el polinomio queda,

$$H_{-1}^{0}(x)=\frac{1}{4}(x-1)^{2}(x+2).$$

2. Para calcular el polinomio $H_{-1}^{1}(x)$ tendremos en cuenta que

$$H_{-1}^{1}(1)=\frac{dH_{-1}^{1}}{dx}(1)=0,$$

con lo que el polinomio tiene la forma

$$H_{-1}^{1}(x)=(x-1)^{2}(a(x+1)+b),$$

por otro lado $H_{-1}^{1}(-1)=0$ y $\frac{dH_{-1}^{1}}{dx}(-1)=1$, y por tanto

$$H_{-1}^{1}(-1)=(-1-1)^{2}(a(-1+1)+b)=4b=0 \quad \rightarrow \quad b=0,$$

$$\frac{dH_{-1}^{1}}{dx}(-1)=2(-1-1)(a(-1+1))+(-1-1)^{2}a=1 \quad \rightarrow \quad a=\frac{1}{4},$$

luego el polinomio nos queda:

$$H_{-1}^{1}(x)=\frac{1}{4}(x-1)^{2}(x+1).$$

3. Para calcular los otros dos polinomios, basta considerar que son funciones simétricas a las dos anteriores y por tanto,

$$H_{1}^{0}(x)=H_{-1}^{0}(-x)=-\frac{1}{4}(x+1)^{2}(x-2),$$
$$H_{1}^{1}(x)=-H_{-1}^{1}(-x)=\frac{1}{4}(x+1)^{2}(x-1).$$

Problema 98 *Calcular la función que interpola, utilizando la función sinc(x) a la función $f(x)=sen(x)$ en los puntos $x_i=-\pi,-\frac{\pi}{2},0,\frac{\pi}{2},\pi$.*

Solución: La interpolación a través de la función $\mathrm{sinc}(x)$ en los puntos $a\cdot i$ viene dada por

$$\tilde{f}(x)\approx\sum_{i=M}^{N}f(x_i)\mathrm{sinc}\left(\pi\left(\frac{x}{a}-i\right)\right)=\sum_{i=M}^{N}f(x_i)\frac{sin\left(\pi\left(\frac{x}{a}-i\right)\right)}{\pi\left(\frac{x}{a}-i\right)},$$

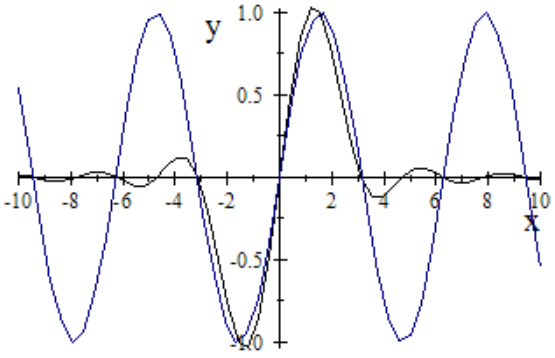
en este caso $x_i=a\cdot i=\frac{\pi}{2}\cdot i$ con $i=-2,-1,0,1,2$, y por tanto obtenemos

$$\tilde{f}(x)\approx f(-\pi)\frac{sin\left(\pi\left(\frac{2x}{\pi}+2\right)\right)}{\pi\left(\frac{2x}{\pi}+2\right)}+f\left(\frac{-\pi}{2}\right)\frac{sin\left(\pi\left(\frac{2x}{\pi}+1\right)\right)}{\pi\left(\frac{2x}{\pi}+1\right)}+f(0)\frac{sin\left(\pi\left(\frac{2x}{\pi}\right)\right)}{\pi\left(\frac{2x}{\pi}\right)}+$$

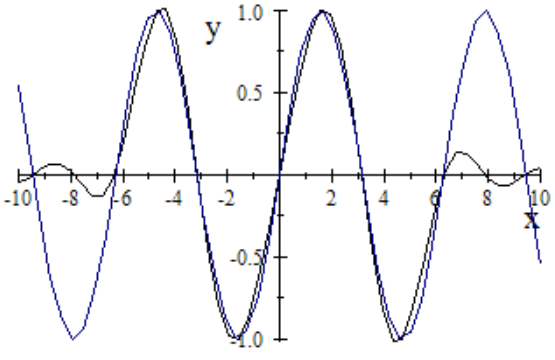
$$+f\left(\frac{\pi}{2}\right)\frac{sin\left(\pi\left(\frac{2x}{\pi}-1\right)\right)}{\pi\left(\frac{2x}{\pi}-1\right)}+f\left(\pi\right)\frac{sin\left(\pi\left(\frac{2x}{\pi}-2\right)\right)}{\pi\left(\frac{2x}{\pi}-2\right)}=$$

$$sin\left(\frac{-\pi}{2}\right)\frac{sin\left(2x+\pi\right)}{2x+\pi}+sin\left(\frac{\pi}{2}\right)\frac{sin\left(2x-\pi\right)}{2x-\pi}=\frac{sin2x}{2x+\pi}-\frac{sin2x}{2x-\pi}=-2\pi\frac{sin2x}{4x^2-\pi^2}.$$

En las siguiente gráficas se muestran el $sen(x)$ y su aproximación usando diferentes formas de interpolación



Comparación del $sen(x)$ con su aproximación numérica utilizando $\mathrm{sinc}(x)$, tomando como puntos de interpolación $x_i = -\pi, \frac{-\pi}{2}, 0, \frac{\pi}{2}, \pi$.



Comparación del $sen(x)$ con su aproximación numérica utilizando $\mathrm{sinc}(x)$, tomando como puntos de interpolación $x_i = -2\pi, \frac{-3\pi}{2}, -\pi, \frac{-\pi}{2}, 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi$

Problema 99 *Calcular el polinomio trigonométrico tomando $N = 2$, que interpola a la función $f(x) = |x|$ en el intervalo $[-\pi, \pi]$.*

Solución: La interpolación por polinomios trigonométricos tiene la forma:

$$\tilde{f}(x) \approx \sum_{k=-2}^{2} c_k e^{ikx},$$

donde los coeficientes se calculan a partir de la siguiente expresión:

$$c_k = \frac{\int_{-\pi}^{\pi} f(x)e^{ikx}dx}{2\pi} = \frac{\int_{-\pi}^{\pi} |x|\, e^{ikx}dx}{2\pi} = \frac{-\int_{-\pi}^{0} xe^{ikx}dx}{2\pi} + \frac{\int_{0}^{\pi} xe^{ikx}dx}{2\pi}.$$

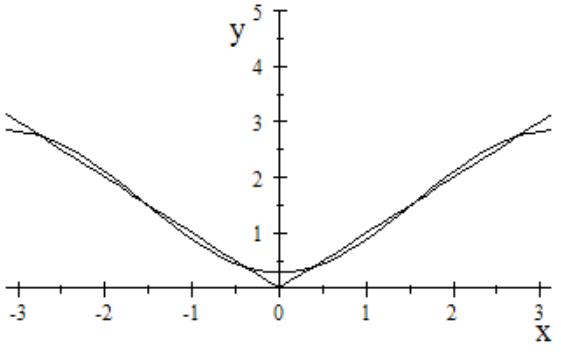
Los valores de estos coeficientes son:

$$c_2 = c_{-2} = 0, \quad c_1 = c_{-1} = \frac{-2}{\pi}, \quad c_0 = \frac{\pi}{2}.$$

Sustituimos en el sumatorio que aproxima a la función y obtenemos:

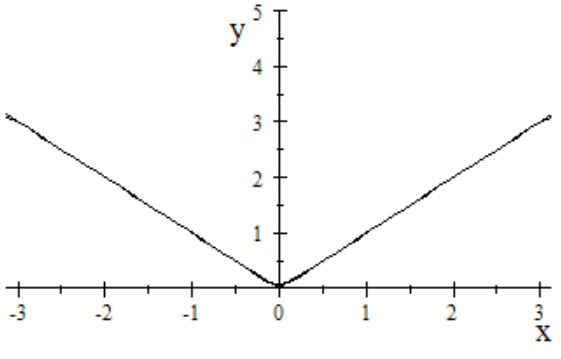
$$\tilde{f}(x) \approx \frac{-2}{\pi}e^{-ix} + \frac{\pi}{2} - \frac{2}{\pi}e^{ix} = \frac{1}{2}\pi - \frac{4}{\pi}\cos x.$$

La siguiente gráfica compara $f(x) = |x|$ con su aproximación $\tilde{f}(x)$ para $N = 2$ en el intervalo $[-\pi, \pi]$.



Polinomio trigonométrico ($N = 2$, $[-\pi, \pi]$)

En la siguiente gráfica se realiza la misma comparación tomando $N = 10$.



Polinomio trigonométrico ($N = 10$, $[-\pi, \pi]$)

BIBLIOGRAFÍA

[ACMM] Alvarez L., Colom M., Morel J.D., Morel J.M. "Computing the daily reproduction number of COVID-19 by inverting the renewal equation", medRxiv 2020.
[Bu-Fa] Burden R., Faires D. "Análisis Numérico", Grupo Editorial Iberoamérica 2000.
[Ci] Ciarlet P.G. "Introduction à l'analyse numérique matricielle et à l'optimisation", Masson , 2007.
[Hi] Higham N. "Accuracy and Stability of Numerical Algorithms", SIAM, 2002
[Hu] Hultquist P. F. "Numerical Methods for Engineers and Computer Scientists", The Benjamin/Cummings Publishing Company, Inc. 1988.
[In-Re] Infante J.A., Rey J.M., "Métodos Numéricos. Teoría, problemas y prácticas con MATLAB". Ediciones Pirámide, 2018.
[Is-Ke] Isaacson E., Keller H. "Analysis of Numerical Methods". John Wiley and Sons, 2003.
[Ki-Ch] Kincaid D., Cheney W. "Análisis Numérico". Addison-Wesley Iberoamericana, 1994.
[La-Th] Lascaux P., Théodor R. "Analyse numérique matricielle appliquée à l'art de l'ingénieur. Vol. 1 Méthodes directes y Vol. 2 Méthodes itératives ", Dunod, 2004.
[Ma] Ma S., Zhang J., Zeng M., Yun Q., Guo W., Zheng Y., Zhao S., Wang M., Yang Z. "Epidemiological parameters of coronavirus disease 2019: a pooled analysis of publicly reported individual data of 1155 cases from seven countries ", medRxiv 2020.
[Prem] Prem K., van Zandvoort K., Klepac P. , Eggo RM. , Davies NG., Cook AR. , Jit M. "Projecting contact matrices in 177 geographical regions: an update and comparison with empirical data for the COVID-19 era". medRxiv 2020.