Matemática Numérica II

Angela León Mecías

Notas de clases, 2020

Índice general

Preface

Estas notas de clases comenzaron a gestarse cuando empecé a digitalizar las Conferencias y Clases Prácticas de la profesora María Victoria Mederos en Septiembre del año 2000, curso en el que me incorporé al colectivo de Matemática Numérica de la Facultad de Matemática y Computación de la Universidad de La Habana. Sueño con que algún día me alcance el tiempo para que se convierta en un libro de texto.

Capítulo 1

Aproximación de funciones por interpolación

¿Por qué y para qué aproximar funciones? La necesidad de buenas técnicas de aproximación surge en variados marcos de la resolución de problemas reales donde no es posible usar funciones dadas por expresiones analíticas exactas, como pueden ser: encontrar la solución de problemas de ecuaciones diferenciales, representar curvas, como la que describe el contorno de la bahía de La Habana, Figura (1.1)[1] o la que modela el contorno de una pieza, Figura (1.2).

Figura 1.1: Bahía de La Habana

En un contexto más matemático un problema tipo puede ser: conocida una función f de forma analítica o de forma discreta por un conjunto de valores, reemplazarla o aproximarla (con un error de aproximación dentro de un rango de tolerancia dado) por una expresión con la cual se pueda operar de forma más simple, a manera de ejemplo, digamos que pueda ser más fácil realizar operaciones tales como diferenciación e integración. Lo más común es que se necesite aproximar por una función un conjunto de valores proveniente de experimentos o mediciones. Algunas de las situaciones que suelen presentarse:

1. Los datos son pocos y se obtienen como resultado de una evaluación precisa de f, es decir, se dispone de valores de una función que son exactos, salvo por errores de redondeo o de truncamiento y nos interesa buscar $f(\overline{x})$ que es desconocida; o puede ser que se conozca la expresión

[1]Tomado de Behar Jequín S., Construcción de una familia de A-splines cúbicos G^2-continuos para la solución de diversos problemas de CAGD. Tesis de doctorado 2008.

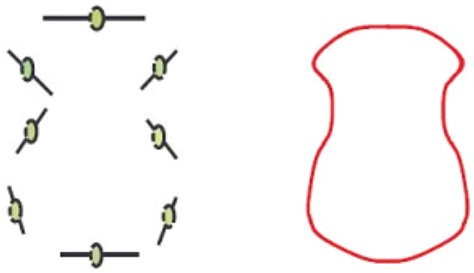


Figura 1.2: Diseño de pieza

analítica de f y esta resulte complicada para hacer operaciones tales como diferenciación o integración. En este caso se usa la **aproximación por interpolación**, ver la Figura 1.3.

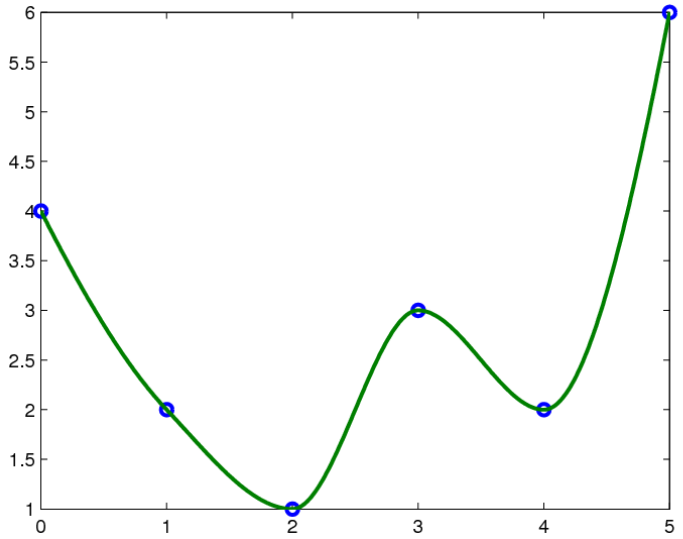


Figura 1.3: Función que interpola los datos

2. Los datos se obtienen a partir de experimentos donde se toman lecturas en tiempos discretos, es decir, en un conjunto de puntos $x_0, x_1, ..., x_m$ se miden valores $f_0, f_1, ...f_m$ de una función $f(x)$, y es de interés obtener una aproximación de f en el punto $\overline{x}$, que no es uno de los tabulados. Este es el caso en que f es una función experimental, cuyos valores medidos están más o menos afectados de error por diversas razones. Aquí se usa la **aproximación mínimo cuadrática**, como se muestra en la Figura 1.4.

3. Se necesita evaluar funciones básicas en una computadora para un $\widetilde{x}$ real arbitrario, con buena precisión, sustituyendo desarrollos en serie por funciones de pocos términos. En este caso se usa la llamada **aproximación uniforme o computer approximation**.

Una de las técnicas más antiguas de aproximación consiste en aproximar una función dada $f(x)$

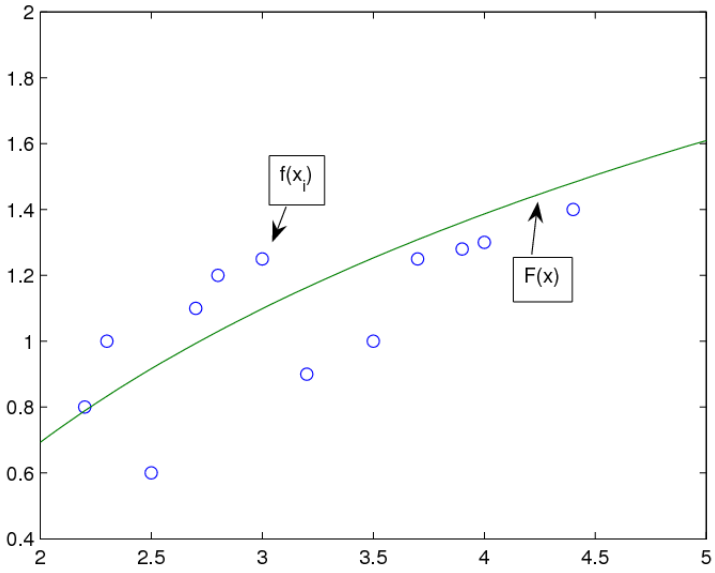


Figura 1.4: Aproximación por mínimos cuadrados

por una suma finita

$$\widetilde{f}(x) = a_1\phi_1(x) + a_2\phi_2(x) + ... + a_n\phi_n(x)$$

de funciones simples $\phi_i(x)$ con buenas propiedades y que sean simples de calcular. Los coeficientes a_i son constantes que se deben determinar a partir de restricciones impuestas a $\widetilde{f}(x)$. Estas ideas se remontan a Fourier[2] en 1822, a Euler[3] y Lagrange[4] en el siglo 18.

Las clases de funciones de aproximación más usadas son: polinomios algebraicos, funciones polinómicas por tramos, funciones trigonométricas, exponenciales y funciones racionales.

Los polinomios son, entre todas, las más usadas, aunque en ocasiones se les señalen deficiencias. Estos son fáciles de evaluar, derivar e integrar. Es importante que la función de aproximación tenga un comportamiento semejante a la función dada, por lo que en muchos casos resulta más natural aproximar por otras funciones, por ejemplo, por funciones trigonométricas si f es periódica. Otra forma de interpretar la aproximación de funciones está relacionada con el hecho de que la mayoría de los problemas matemáticos se pueden representar mediante una ecuación operacional

$$Ax = y \tag{1.1}$$

donde las variables independientes están representadas por x, las dependientes por y y A es un operador o función.

En este contexto, la determinación del operador A, que se conoce como un problema de **identificación de parámetros**, es precisamente un problema de aproximación de funciones. Por ejemplo,

[2]Jean Baptiste Joseph Fourier, matemático (21 de marzo de 1768 en Auxerre, Bourgogne, Francia-París 16 de mayo de 1830).

[3]Leonard Euler, matemático y físico (15 de abril de 1707, Basilea, Suiza-18 de septiembre de 1783, San Petersburgo, Rusia).

[4]Joseph Louis Lagrange, físico, matemático y astrónomo (25 de enero de 1736, Turín, Italia-10 de abril de 1813, París, Francia).

se tiene una tabla de valores que representan una cierta función f, cuya expresión analítica se desconoce. Por otra parte, la solución de sistemas de ecuaciones lineales donde x es la incógnita, se considera un problema inverso, y en el caso en que y es la incógnita, estamos ante un problema directo, como lo es el cálculo de la integral de una función dada; problema que se abordará más adelante, precisamente como una aplicación de la aproximación de funciones. En lo que sigue, se tratará la aproximación por interpolación y la aproximación mínimo cuadrática con diferentes bases.

1.1. Interpolación

La interpolación es una de las aplicaciones más antiguas de la Matemática Numérica, en la que se apoyan otros algoritmos numéricos como la derivación y la integración numérica.

Desde que el hombre comenzó a diseñar, digamos, por ejemplo, el casco de los barcos alrededor de 1800, se presentó el problema de cómo dibujar (manualmente) una curva suave que pasara por un conjunto de puntos dados. Una forma de solucionar el problema fue poniendo pesos de metal ("ducks") en los puntos dados y luego pasar una barra de madera elástica (llamada *spline*) entre los pesos. Este mismo principio es usado en nuestros días cuando no existe un programa de diseño apropiado o para verificar manualmente los resultados computacionales. La interpolación se encuentra entre las bases matemáticas del diseño geométrico asistido por computadoras, conocido por sus siglas en inglés CAGD (Computer Aided Geometric Design), el diseño de letras en los lenguajes gráficos tales como PostScript, entre otros. Una función de interpolación es aquella que pasa a través de puntos dados como datos, los cuales se muestran comúnmente por medio de una tabla de valores o se toman directamente de una función dada.

Dada $f(x) \in C[a,b]$ por $n+1$ pares de puntos (n una cantidad pequeña): $(x_0, f_0), (x_1, f_1), ..., (x_n, f_n)$, para los cuales los valores de f han sido calculados con una buena precisión. Estamos entonces ante un problema de interpolación si para la función de aproximación que se busca $F(x_i, a_0, ..., a_n)$, que depende de $n+1$ parámetros a_i, estos deben ser determinados de forma tal que para los $n+1$ pares de números reales ó complejos

$$(x_i, f_i), i = 0, 1, ..., n \quad x_i \neq x_k \text{ para } i \neq k$$

se cumple

$$F(x_i, a_0, ..., a_n) = f(x_i) = f_i \quad i = 0, 1, ..., n$$

A la condición anterior se le llama condición de interpolación. *El problema lineal de interpolación* consiste en aproximar la función $f \in C[a,b]$ por $F \in \Phi \subset C[a,b]$ mediante una combinación lineal de $n+1$ funciones φ_j que constituyan una base prefijada de Φ

$$F(x) = \sum_{j=0}^{n} a_j \varphi_j(x) \tag{1.2}$$

y tal que F satisfaga la condición de interpolación. Los puntos distintos x_i se denominan nodos de interpolación:

Casos particulares de la interpolación lineal

- interpolación polinomial

$$F(x_i, a_0, ..., a_n) \equiv a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$$

- interpolación trigonométrica

$$F\left(x_i, a_0, ..., a_n\right) \equiv a_0 + a_1 e^{xi} + a_2 e^{2xi} + ... + a_n e^{nxi} \qquad \left(i^2 = -1\right)$$

- interpolación por splines: los splines son funciones polinómicas definidas por tramos, a las que se exige determinado grado de derivabilidad, donde el orden de los polinomios en cada tramo depende del orden del spline, p.e., para un spline cúbico estaremos hablando de polinomios cúbicos por tramos.

Casos particulares de la interpolación no lineal

- interpolación por funciones racionales

$$F\left(x_i; a_0, ..., a_n, b_0, ..., b_m\right) \equiv \frac{a_0 + a_1 x + a_2 x^2 + ... + a_n x^n}{b_0 + b_1 x + b_2 x^2 + ... + b_m x^m}$$

- interpolación a través de sumas exponenciales

$$F\left(x_i; a_0, ..., a_n, \lambda_0, ..., \lambda_n\right) \equiv a_0 e^{\lambda_0 x} + a_1 e^{\lambda_1 x} + a_2 e^{\lambda_2 x} + ... + a_n e^{\lambda_n x}$$

1.1.1. Interpolación polinomial. Fórmula de Lagrange

Si consideramos la expresión (1.2) en los $n+1$ puntos distintos x_i, obtenemos un sistema lineal de $n+1$ ecuaciones con $n+1$ incógnitas $a_0, a_1, \ldots, a_n$

$$\sum_{j=0}^{n} a_j \varphi_j(x_i) = f(x_i) \; i = 0, .., n \tag{1.3}$$

$$\begin{array}{rcll} i & = & 0, & a_0\varphi_0\left(x_0\right) + a_1\varphi_1\left(x_0\right) + \ldots + a_n\varphi_n\left(x_0\right) = f\left(x_0\right) \\ i & = & 1, & a_0\varphi_0\left(x_1\right) + a_1\varphi_1\left(x_1\right) + \ldots + a_n\varphi_n\left(x_1\right) = f\left(x_1\right) \\ & & \vdots & \\ i & = & n, & a_0\varphi_0\left(x_n\right) + a_1\varphi_1\left(x_n\right) + \ldots + a_n\varphi_n\left(x_n\right) = f\left(x_n\right) \end{array} \tag{1.4}$$

Este sistema tendrá solución única si y solo si

$$\det \begin{bmatrix} \varphi_0\left(x_0\right) & \varphi_1\left(x_0\right) & \ldots & \varphi_n\left(x_0\right) \\ \varphi_0\left(x_1\right) & \varphi_1\left(x_1\right) & \ldots & \varphi_n\left(x_1\right) \\ \vdots & \vdots & \ldots & \vdots \\ \varphi_0\left(x_n\right) & \varphi_1\left(x_n\right) & \ldots & \varphi_n\left(x_n\right) \end{bmatrix} \neq 0 \tag{1.5}$$

Observe que el valor del determinante depende, tanto de las funciones base $\varphi_j\left(x\right)$ como de los nodos.

Si $F\left(x\right)$ es un polinomio de grado menor o igual que n, $F \in \Phi = P_n\left[a, b\right]$, entonces

$$p_n\left(x\right) = F\left(x\right)$$

es un polinomio de interpolación.

De forma natural, nos hacemos inmediatamente algunas preguntas: ¿tiene solución el problema de interpolación?, ¿es única?, ¿qué se puede decir del error de interpolación?

Teorema 1 *Dados $n+1$ puntos distintos*

$$(x_i, f_i), i = 0, ...n \quad x_i \neq x_k \quad para\ i \neq k$$

existe un único polinomio de grado menor ó igual que n, $p_n(x) \in \Phi$ tal que

$$p_n(x_i) = f(x_i) \tag{1.6}$$

Demostración 2 *Unicidad: supongamos que existen dos polinomios $p(x)$ y $q(x)$ de grado menor o igual que n que cumplen la exigencia de interpolación (1.6). Entonces $r(x) = p(x) - q(x)$ es de grado menor o igual que n y $r(x_i) = p(x_i) - q(x_i) = f(x_i) - f(x_i) = 0, 0 \leq i \leq n$ se anula en $n+1$ puntos lo que significa que $r(x)$ tiene $n+1$ raíces, lo cual es una contradicción, de aquí que $r(x) \equiv 0$ y, por tanto, $p(x) = q(x)$ en contradicción con la hipótesis.*

Existencia: demostrar que se puede construir un polinomio que satisface (1.6). Si se escoge como base el conjunto $\{\varphi_j(x)\}^n = \{1, x, x^2, ..., x^n\}$, entonces el polinomio de grado n puede escribirse

$$p_n(x) = \sum_{j=0}^{n} a_j \varphi_j(x) = \sum_{j=0}^{n} a_j x^j \tag{1.7}$$

e imponiendo la exigencia de interpolación, se obtiene

$$p_n(x_i) = \sum_{j=0}^{n} a_j x_i^j = f(x_i), \quad 0 \leq i \leq n \tag{1.8}$$

lo que representa el siguiente sistema de ecuaciones lineales para la determinación de los coeficientes a_i:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \tag{1.9}$$

Esto es $Va = f$. Si los nodos de interpolación son todos distintos, entonces el determinante de la matriz V es distinto de 0, pues se trata del determinante de Vandermonde y se puede escribir de la forma

$$\det(V) = \prod_{k=1}^{n} (x_k - x_j), \ k > j$$

lo que garantiza la existencia de la solución del sistema, ya que entonces,

$$rango(V) = rango(V, f)$$

y con ello la existencia del polinomio de interpolación de grado $\leq n$.

Polinomio de interpolación de Lagrange

El cálculo de los coeficientes a_j del polinomio de interpolación se puede realizar resolviendo el sistema (1.9), que resulta de usar la base

$$\left\{1, x, x^2, ..., x^n\right\}. \tag{1.10}$$

Sin embargo, en muchos casos (en dependencia de n y de los nodos x_i), la matriz V resulta muy mal condicionada. Supongamos, por ejemplo, que los nodos x_i son equidistantes en el intervalo $[0,1]$, entonces las sucesivas potencias de $1, x, x^2, ..., x^n$ son casi linealmente dependientes sobre el intervalo $[0,1]$. Como se observa en la Figura (1.5), las funciones de esta base son todas positivas en el intervalo dado y toman valores partiendo del punto $(0,0)$ hasta el $(1,1)$ cuando $n > 0$:

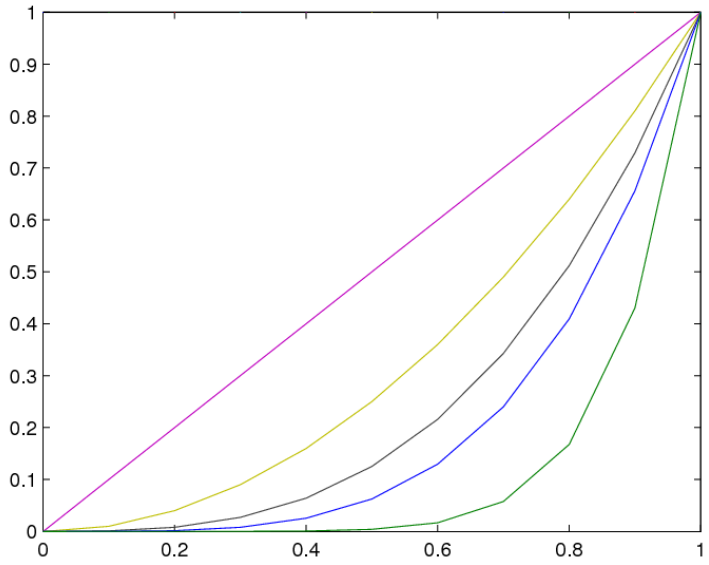


Figura 1.5: Funciones de la base $\{1, x, x^2, ..., x^n\}$

Esta casi dependencia lineal de las columnas de V es lo que dificulta la resolución del sistema $Va = f$ con precisión simple para $n > 6$ o $n > 7$.

Veamos ahora cómo calcular el polinomio de interpolación en forma mucho más satisfactoria, sin necesidad de resolver el sistema $Va = f$ de $n+1$ ecuaciones e incógnitas. Consideremos para ello en lugar de la base (1.10), otra base de $\Phi = P_n[a,b]$: el conjunto $\{l_j(x)\}_{j=0}^n$ de polinomios de grado n de Lagrange.

Sean los $l_j(x)$, polinomios de grado n de la forma

$$l_j(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_{j-1})(x-x_{j+1})\dots(x-x_n)}{(x_j-x_0)(x_j-x_1)\dots(x_j-x_{j-1})(x_j-x_{j+1})\dots(x_j-x_n)}, \tag{1.11}$$

que cumplen

$$l_j(x_i) = \begin{cases} 1, & \text{si} \quad i = j \\ 0, & \text{si} \quad i \neq j. \end{cases} \tag{1.12}$$

Los polinomios $l_j(x)$ que constituyen la llamada base de Lagrange (queda de ejercicio al lector demostrar que los $l_j(x)$ constituyen una base del espacio de los polinomios) tienen la siguiente representación gráfica: Las funciones $l_j(x)$ de la base de Lagrange son marcadamente distintas para cada j, (se puede demostrar que son ortogonales, respecto al producto escalar $\langle f, g\rangle := \int_a^b f(x)g(x)dx$, es decir, se verifica $\langle l_j, l_k\rangle = 0$, $j \neq k$). Al usar la base de Lagrange, el polinomio de interpolación

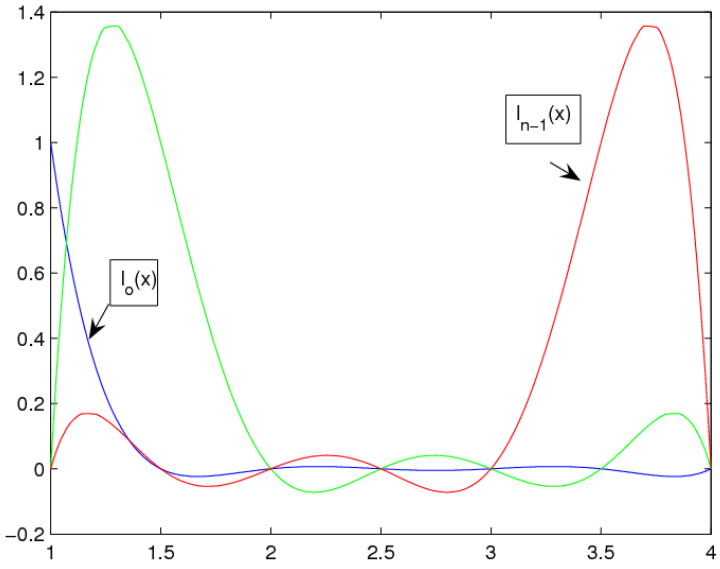


Figura 1.6: Funciones base de Lagrange

tendrá la forma

$$p_n(x) = \sum_{j=0}^{n} a_j l_j(x) \tag{1.13}$$

y la exigencia de interpolación

$$p_n(x_i) = f(x_i), \quad 0 \le i \le n,$$

trae como consecuencia que

$$\sum_{j=0}^{n} a_j l_j(x_i) = f(x_i),$$

de donde, teniendo en cuenta las propiedades de las funciones l_j, para $i = j$,

$$a_j l_j(x_j) = f(x_j) \Rightarrow a_j = f(x_j), \quad 0 \le j \le n. \tag{1.14}$$

Luego sustituyendo (1.14) en (1.13), se obtiene la **fórmula de Lagrange** para el polinomio de interpolación de grado $\le n$ con nodos $x_0, x_1, ..., x_n$

$$\varphi(x) = p_n(x) = \sum_{j=0}^{n} f(x_j) l_j(x), \quad l_j(x) = \prod_{\substack{i=0 \\ i \ne j}} \left(\frac{x - x_i}{x_j - x_i} \right). \tag{1.15}$$

El uso de la base de Lagrange ha permitido obtener los coeficientes a_j del polinomio sin tener que resolver el sistema (1.4), ya que la matriz de elementos $\varphi_{ij} = \varphi_i(x_j)$ es la identidad.

Existen otras muchas formas de expresar el polinomio de interpolación, algunas de las cuales consideramos a continuación.

La utilidad del cambio de base para la forma de Lagrange fue facilitar el cálculo de la función interpolante. En otros casos, el cambio de base puede perseguir como objetivo el dar una visión especial sobre la función interpolante, como cuando se usa la base de Bernstein $\{B_i^n(x)\}$:

$$B_i^n(x) = \begin{pmatrix} n \\ i \end{pmatrix} \frac{(b-x)^{n-i}(x-a)^i}{(b-a)}, \qquad a \le x \le b$$

para el diseño geométrico asistido por computadora, para más detalles (ver Kahaner).

Ejemplo 3 *Construir el polinomio de interpolación de Lagrange a partir de los siguientes datos:*

x	0	2	3	5
$f(x)$	1	3	2	5

$$\varphi(x) \equiv p_3(x) = \sum_{i=0}^{3} f(x_j)\, l_j(x) \tag{1.16}$$

$$l_0(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}$$

$$l_1(x) = \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}$$

$$l_2(x) = \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}$$

$$l_3(x) = \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}$$

Sustituyendo en (1.16) y agrupando en potencias de x obtenemos:

$\varphi(x) = \frac{3}{10}x^3 - \frac{13}{6}x^2 + \frac{62}{15}x + 1,$

1. Si ahora se quiere calcular $\varphi(x) = p_2(x)$, habría que hacer todos los cálculos desde el principio, sin que se puedan aprovechar los cálculos efectuados.

2. El polinomio de interpolación

$$p_n(x) \equiv F(x) = \sum_{j=0}^{n} a_j \varphi_j(x), \qquad \varphi = \{\varphi_j(x)\}_{j=0}^{n}$$

es único, pero se puede representar de diversas formas:

- $\varphi_j(x) = x^j$, con $n = 2$,

$$\begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \end{bmatrix}$$

Hay que resolver el sistema con matriz de Vandermonde para obtener $p_2(x) = a_0 + a_1 x + a_2 x^2$.

- $\varphi_j(x) = l_j(x)$, forma de Lagrange (no hay que resolver un sistema pero hay que prefijar n)

  $a_j = f(x_j)$

$$P_2(x) = f(x_0)\, l_0(x) + f(x_1)\, l_1(x) + f(x_2)\, l_2(x).$$

- forma de Horner, multiplicación anidada o Ruffini (para evaluar con mínimo de operaciones)

$$p_2(x) = (a_1 + a_2 x)\, x + a_0.$$

- forma de Bernstein (para diseño geométrico asistido por computadora)

$$\varphi_j(x) = B_j^n(x) = \begin{pmatrix} n \\ j \end{pmatrix} \frac{(b-x)^{n-j}(x-a)^j}{(b-a)^n}, \quad x \in (a,b)$$

$$p_2(x) = a_0 \begin{pmatrix} 2 \\ 0 \end{pmatrix} \frac{(b-x)^2}{(b-a)^2} + a_1 \begin{pmatrix} 2 \\ 1 \end{pmatrix} \frac{(b-x)(x-a)}{(b-a)^2} + a_2 \begin{pmatrix} 2 \\ 2 \end{pmatrix} \frac{(x-a)^2}{(b-a)^2}.$$

- forma de Newton en diferencias divididas (para evaluar con máxima precisión, en un punto $\widetilde{x}$).
- forma de Newton con nodos equidistantes (para derivar e integrar numéricamente).
- forma de Newton con diferencias finitas retrógradas (para resolver el problema de Cauchy en ecuaciones diferenciales ordinarias).
- forma de Hermite: $p_2(x_i) = f(x_i), p_2'(x_i) = f'(x_i)$ (cuando se conoce una tabla de valores de x, f, f').

Inconvenientes de la fórmula de Lagrange

En la práctica, no siempre se conoce *a priori* cuántos nodos de interpolación deben usarse para lograr una cierta precisión de $p_n(\overline{x})$. Entonces, si denotamos por $p_i(x)$ el polinomio de grado menor o igual que i que interpola a $f(x)$ en los puntos $x_0, ..., x_i$, podemos calcular los polinomios $p_0(x), p_1(x), p_2(x), ...$, incrementando el número de nodos y con ello el grado del polinomio de interpolación, esperando obtener así una aproximación $p_n(x)$ de $f(x)$ satisfactoria.

En un proceso tal, el uso de la fórmula de Lagrange para el polinomio de interpolación $p_n(x)$ no es conveniente, pues no permite aprovechar los cálculos realizados para determinar $p_{n-1}(x)$ y habría que comenzar otra vez como si no se hubiera hecho nada. Esta desventaja de los cálculos en la forma de Lagrange puede aliviarse utilizando el esquema de Aitken, (página 50 del Conte), para el cual, sin embargo, la complejidad computacional es mayor.

Veamos ahora cómo construir el polinomio de interpolación de grado n con nodos $x_0, x_1, ..., x_n$, aprovechando los cálculos ya efectuados para determinar $p_{n-1}(x)$ con nodos $x_0, x_1, ..., x_{n-1}$:

$$p_n(x) = p_{n-1}(x) + h(x)$$

donde $h(x)$ deberá ser un polinomio de grado n para que $p_n(x)$ lo sea. Esto da lugar a la llamada fórmula de Newton.

1.1.2. La fórmula de interpolación de Newton

- Incoveniencia de la fórmula de Lagrange para calcular valores interpolados con una precisión prefijada.
- ¿Cómo resuelve este problema la fórmula de Newton?
- ¿Cómo calcular eficientemente valores interpolados con la precisión deseada mediante el algoritmo de interpolación con número creciente de nodos y el uso de una tabla de diferencias divididas?

Como se vio, el uso de la fórmula de Lagrange requiere prefijar el grado del polinomio y hay situaciones en las que a priori esto no se puede hacer. Por ejemplo, cuando lo que se quiere es obtener un valor interpolado con precisión prefijada y no se conoce cuál es el grado del polinomio de interpolación con el cual se logra $p_n(x^*) \approx f(x^*)$ con esa precisión.

Para ello, puede construirse la sucesión de polinomios $p_0(x), p_1(x), p_2(x), ...$ y paralelamente la sucesión de valores interpolados $p_0(x^*), p_1(x^*), p_2(x^*), ...$ incrementando el número de nodos y, por consiguiente, el grado del polinomio, esperando obtener así la aproximación $p_n(x^*)$ de $f(x^*)$ deseada.

Este proceso requiere que en la obtención de $p_n(x)$ se puedan aprovechar los cálculos efectuados para la obtención de $p_{n-1}(x)$ con nodos $x_0, ..., x_{n-1}$, añadiendo el nuevo nodo x_n

$$p_n(x) = p_{n-1}(x) + h(x)$$

donde $h(x)$ deberá ser un polinomio de grado n.

Diferencias divididas (o cocientes de diferencias)

Las diferencias divididas juegan el papel de herramienta auxiliar para el uso de la fórmula de Newton en el polinomio de interpolación.

Si la función f es continua en un cierto intervalo $[x_0, x_0 + h]$, se define la derivada de f en el punto x_0 como:

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

si este límite existe. Cuando el proceso de paso al límite no se realiza, la expresión

$$\frac{f(x_0 + h) - f(x_0)}{(x_0 + h) - x_0} = \frac{f(x_1) - f(x_0)}{h}, \text{ donde } x_1 = x_0 + h$$

se denomina **primera diferencia dividida** de f con respecto a x_0 y x_1, y se denota por $f[x_0, x_1]$:

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f(x_0)}{x_0 - x_1} + \frac{f(x_1)}{x_1 - x_0}$$

Repitiendo el proceso, obtenemos la segunda diferencia dividida de f con respecto a x_0, x_1, x_2 como diferencia dividida de las primeras diferencias divididas $f[x_1, x_2]$ y $f[x_0, x_1]$:

$$\begin{aligned} f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\ &= \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{f(x_1)}{(x_1 - x_0)(x_1 - x_2)} + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)} \end{aligned}$$

y así sucesivamente, la n-ésima diferencia dividida de f con respecto a $x_0, x_1, x_2, ..., x_n$ será la diferencia dividida de las $(n-1)$-ésimas diferencias divididas $f[x_1, .., x_n]$ y $f[x_0, .., x_{n-1}]$:

$$f[x_0, .., x_n] = \frac{f[x_1, .., x_n] - f[x_0, .., x_{n-1}]}{x_n - x_0} \tag{1.17}$$
$$= \sum_{i=0}^{n} \frac{f(x_i)}{(x_i - x_0) ... (x_i - x_{i-1}) (x_i - x_{i+1}) ... (x_i - x_n)}$$

La tabla de diferencias divididas

Para construir una tabla de diferencias divididas nos basamos en la expresión recurrente

$$f[x_j, .., x_{j+k}] = \frac{f[x_{j+1}, .., x_{j+k}] - f[x_j, .., x_{j+k-1}]}{x_{j+k} - x_j} \tag{1.18}$$

que nos permite generar todas las diferencias divididas.

En particular, para $(j = 0, k = i)$:

$$f[x_0, .., x_i] = \frac{f[x_1, .., x_i] - f[x_0, .., x_{i-1}]}{x_i - x_0}$$

que son las diferencias que encabezan cada columna de la tabla. El cálculo de la tabla se efectúa por columnas, usando la fórmula recursiva (1.18) para $k = 1, 2, ...$, es decir, primero se calculan todas las primeras diferencias divididas, después todas las segundas diferencias divididas, y así sucesivamente

$$\begin{array}{llllll} x_0 & f[x_0] & & & & \\ x_1 & f[x_1] & f[x_0, x_1] & & & \\ x_2 & f[x_2] & f[x_1, x_2] & f[x_0, x_1, x_2] & & \\ & f[x_3] & f[x_2, x_3] & f[x_1, x_2, x_3] & f[x_0, x_1, x_2, x_3] & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{array}$$

Para automatizar el cálculo de la tabla, es necesario racionalizar el almacenamiento de sus valores, pues si se usara un arreglo bidimensional, desperdiciaría casi la mitad del mismo debido a la forma triangular de la tabla. Bastaría guardar en un arreglo unidimensional los elementos de la tabla que se usan al evaluar cada sumando de la fórmula de Newton, es decir, las diferencias divididas $f[x_1, \ldots, x_i]$, $0 \leq i \leq n$, esto se logra calculando los valores de la tabla, no por columna, sino por fila, colocando los nuevos valores sobre los que ya se usaron.

Algoritmo 4

Dados los $n+1$ nodos distintos $x_0, x_1, \ldots, x_n$
y los valores correspondientes $f(x_0), \ldots, f(x_n)$ almacenados en d_i $(0 \leq i \leq n)$:
Para $k = 0, 1, 2, \ldots, n$ (k orden de la diferencia dividida)
Para $j = k, k-1, \ldots, 0$ (j : nodo inicial)
calcular $\frac{f[x_{j+1}, ..., x_{k+1}] - f[x_j, ..., x_k]}{x_{k+1} - x_j} = f[x_j, \ldots, x_{k+1}]$
mediante $\frac{d_{j+1} - d_j}{x_{k+1} - x_j} \to d_j$

Deducción de la fórmula de Newton[5]

Se considera

$$p_n(x) = p_{n-1}(x) + h(x) \tag{1.19}$$

[5] 1643-1727

donde $p_{n-1}(x)$ es el polinomio de interpolación ya calculado correspondiente a los nodos $x_0, ..., x_n$, y $h(x)$ cumple las siguientes propiedades:

- $h(x)$ tiene que ser un polinomio de grado n.
- $h(x_i) = p_n(x_i) - p_{n-1}(x_i) = f(x_i) - f(x_i) = 0, 0 \leq i \leq n-1$, luego los n nodos $x_0, ..., x_{n-1}$ son los n ceros de $h(x)$.

De ahí que $h(x)$ pueda expresarse en la forma

$$h(x) = a_n (x - x_0)(x - x_1) ... (x - x_{n-1}), \tag{1.20}$$

donde a_n es una constante a determinar. Sustituyendo (1.20) en (1.19), obtenemos:

$$p_n(x) = p_{n-1}(x) + a_n (x - x_0)(x - x_1) \dots (x - x_{n-1}). \tag{1.21}$$

Para determinar la constante a_n, exigimos que se cumpla la condición de interpolación en el nuevo nodo x_n

$$p_n(x_n) = f(x_n)$$

entonces

$$p_{n-1}(x_n) + a_n (x_n - x_0)(x_n - x_1) \dots (x_n - x_{n-1}) = f(x_n),$$

de donde

$$a_n = \frac{f(x_n) - p_{n-1}(x_n)}{(x_n - x_0) \dots (x_n - x_{n-1})}. \tag{1.22}$$

Considerando la fórmula de Lagrange para $p_{n-1}(x)$

$$p_{n-1}(x) = \sum_{i=0}^{n-1} f(x_i) l_i(x) = \sum_{i=0}^{n-1} f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^{n-1} \left(\frac{x - x_j}{x_i - x_j} \right),$$

y evaluando $p_{n-1}(x_n)$ en $x = x_n$, y sustituyendo en (1.22) y simplificando se obtiene

$$a_n = \sum_{i=0}^{n} \frac{f(x_i)}{(x_i - x_0) ..(x_i - x_{i-1})(x_i - x_{i+1})...(x_i - x_n)}. \tag{1.23}$$

Comparando (1.23) con la expresión de la n-ésima diferencia dividida de f con respecto a los puntos $x_0, x_1, ..., x_n$, llegamos a que

$$a_n = f[x_0, x_1, ..., x_n], \tag{1.24}$$

y sustituyendo (1.24) en (1.21), obtenemos

$$p_n(x) = p_{n-1}(x) + f[x_0, x_1, ..., x_n](x - x_0)(x - x_1) ... (x - x_{n-1}). \tag{1.25}$$

Como $p_{n-1}(x)$ es de grado menor que n, y $(x - x_0)(x - x_1) ... (x - x_{n-1}) = x^n +$ un polinomio de grado menor que n, entonces, $p_n(x) = f[x_0, x_1, ..., x_n] x^n +$ un polinomio de grado menor que n.

Para $n = 0$,

$$p_0(x) = f[x_0] x^0 = f[x_0]$$

y como por la exigencia de interpolación, $p_0(x_0) = f(x_0)$, entonces

$$f[x_0] = f(x_0), \tag{1.26}$$

lo que podemos tomar como definición de la diferencia dividida de orden 0 (cero) de f en x_0. Teniendo en cuenta (1.25) y (1.26), obtenemos finalmente

$$\begin{aligned} p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ + \cdots + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1)\ldots(x - x_{n-1}), \end{aligned} \tag{1.27}$$

que es la fórmula de Newton para el polinomio de interpolación de grado n y puede escribirse en forma compacta como

$$p_n(x) = \sum_{i=0}^{n} f[x_0, \ldots, x_i] \prod_{j=0}^{i-1} (x - x_j). \tag{1.28}$$

Se nota la necesidad de calcular en cada sumando la diferencia dividida de orden i de f con respecto a los nodos $x_0, x_1, ..., x_i$.

Observación 5 *Ahora la base es*

$$\{\varphi_j(x)\}_{i=0}^{n} = \{1, (x - x_0), (x - x_0)(x - x_1), ..., (x - x_0)(x - x_1)...(x - x_{n-1})\}$$

Observación 6 *La forma de Newton puede evaluarse eficientemente mediante el algoritmo de multiplicación anidada para la forma de Newton, que requiere un total de 3n operaciones en punto flotante a diferencia de las 4n operaciones que requiere la forma de Lagrange.*

Ejemplo 7 *Dada la siguiente tabla de la función f, halle aproximaciones de $f(1)$ a partir de $p_n(1)$ para $n = 0, 1, 2, 3$.*

x	0	2	3	5
$f(x)$	1	3	2	5

Teniendo en cuenta la fórmula de Newton (1.28), vemos que para $n = 3$ es necesario usar las diferencias divididas $f[x_0, x_1]$, $f[x_0, x_1, x_2]$ y $f[x_0, x_1, x_3]$. Construyamos para ello una tabla de diferencias divididas a partir de $x_0 = 0$, de modo que el intervalo $[x_0, x_1]$ contenga al punto $x^* = 1$ donde se desea interpolar

n	x	$f(x)$	$f[,]$	$f[,.,]$	$f[,.,.,]$
0	0	1			
1	2	3	1		
2	3	2	-1	$-\frac{2}{3}$	
3	5	5	$\frac{3}{2}$	$\frac{5}{6}$	$\frac{3}{10}$

Se construyen las aproximaciones de $f(1)$ como sigue:

$n = 0$: $p_0(x^*) = f(x_0) = 1$

$n = 1$: $p_1(x^*) = p_0(x^*) + f[x_0, x_1](x^* - x_0)$

$p_1(1) = 2$

$n = 2$: $p_2(x^*) = p_1(x^*) + f[x_0, x_1, x_2](x - x_0)$,1. $(x^* - x_1)$

$$\begin{array}{rl} & p_2(1) = 2{,}67 \\ n=3: & p_3(x^*) = p_2(x^*) + f[x_0, x_1, x_2, x_3](-1)(x^* - x_2) \\ & p_3(1) = 3{,}27 \end{array}$$

La sucesión obtenida de aproximaciones de $f(1)$ *es:*

$$p_0(1) = 1, \quad p_1(1) = 2, \quad p_2(1) = 2{,}67, \quad p_3(1) = 3{,}27$$

¿Cuál es la mejor de las cuatro?. ¿Es la sucesión de aproximaciones convergente?. ¿Convendrá tomar n mayor?

Estas interrogantes se responderán más adelante.

Teorema 8 *La diferencia dividida* $f[x_0, \ldots, x_k]$ *es una función simétrica de los* x_i*. Es* $x_{i0}, \ldots, x_{ik}$ *una permutación de los números* $x_0, \ldots, x_k$*, entonces se cumple*

$$f[x_{i0}, \ldots, x_{ik}] = f[x_0, \ldots, x_k]$$

Demostración 9 *Según (1.28)* $f[x_0, \ldots, x_k]$ *es el coeficiente de la mayor potencia del polinomio de interpolación* $P_{0,\ldots,k}$ *que pasa por los puntos* $(x_i, f_i), i = 0, \ldots, k$*. Teniendo en cuenta que el polinomio de interpolación es único entonces se cumple* $P_{i_0,\ldots,i_k}(x) \equiv P_{0,\ldots,k}(x)$ *para una permutación cualquiera* $i_0, \ldots, i_k$ *de los números* $0, 1, \ldots, k$*. En particular se cumple entonces que*

$$f[x_{i0}, \ldots, x_{ik}] = f[x_0, \ldots, x_k]$$

Interpolación con número creciente de nodos

Consideremos ahora el problema de estimar el valor $f(x^*), (x^* \neq x_i)$ con precisión prefijada, utilizando un número creciente de nodos para el polinomio de interpolación.

Partimos de $p_0(x^*) = f(x_0)$ y calculamos $p_1(x^*)$ con nodos x_0, x_1; $p_2(x^*)$ con nodos x_0, x_1, x_2; y así sucesivamente $p_{n-1}(x^*)$ y $p_n(x^*)$, con la esperanza de que la diferencia entre los valores interpolados $p_{n-1}(x^*)$ y $p_n(x^*)$ se haga suficientemente pequeña

$$|p_n(x^*) - p_{n-1}(x^*)| < \varepsilon$$

Como planteamos anteriormente, la fórmula de Newton está expresamente diseñada para este fin. Pero puede suceder:

- que se acaben los nodos.
- que $|p_n(x^*) - p_{n-1}(x^*)|$ comience a crecer.

Si se terminan los nodos y la secesión de las diferencias $|p_n(x^*) - p_{n-1}(x^*)|$ es decreciente al crecer n, el último valor interpolado calculado $p_n(x^*)$ será la mejor aproximación de $f(x^*)$, aunque no se haya alcanzado la precisión deseada.

Si $|p_n(x^*) - p_{n-1}(x^*)|$ empieza a crecer para un cierto n, ello es indicación de que la precisión del valor interpolado no aumenta al incluir un nodo más, y entonces el penúltimo valor interpolado calculado $p_{n-1}(x^*)$ será la mejor aproximación de $f(x^*)$. Esta situación pone en evidencia que el polinomio de interpolación de grado n no necesariamente converge a la función continua f para

todo $x^* \in [x_0, x_n]$ cuando $n \to \infty$, dado el conjunto de nodos x_i $(0 \leq i \leq n)$. Un buen ejemplo ilustrativo es la función de Runge (1901);

$$f(x) = \frac{1}{1+25x^2}, \quad -1 \leq x \leq 1$$

Runge intentó aproximar esta función por polinomios de interpolación con nodos equidistantes, y descubrió que cuando $n \to \infty$, $p_n(x)$ diverge en los intervalos $0{,}726 \leq |x| < 1$, mientras que en la parte central del intervalo $[-1, 1]$ la aproximación es satisfactoria, ver Figura (1.7).

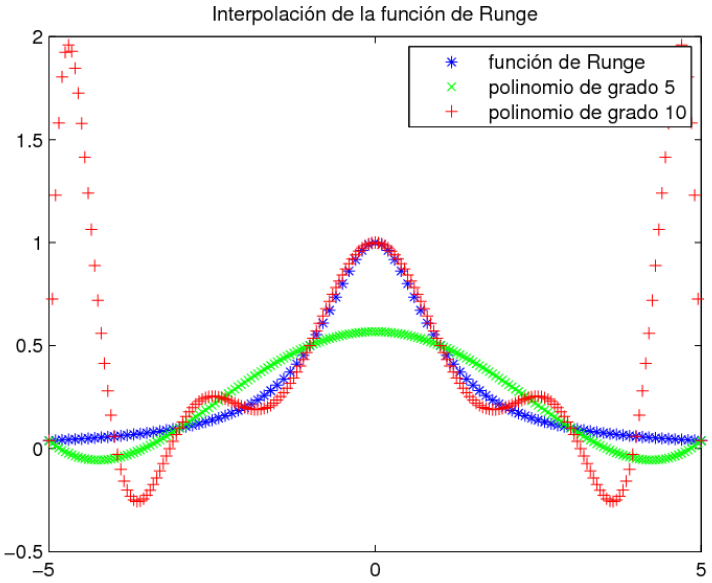


Figura 1.7: Función de Runge

Denotemos por $\psi_i(x)$ la productoria de la función de Newton:

$$\psi_i(x) = \prod_{j=0}^{i-1}(x - x_j) = (x - x_0)\ldots(x - x_{i-1}) \tag{1.29}$$

Si conocemos los valores $p_{n-1}(x^*), \psi_{n-1}(x^*)$ y $f[x_0, \ldots, x_n]$, entonces podemos calcular la nueva aproximación del valor interpolado $p_n(\overline{x})$ mediante la expresión

$$p_n(x^*) = p_{n-1}(x^*) + f[x_0, .., x_n]\,\psi_{n-1}(x^*)(x^* - x_{n-1})$$

Datos: nodos $x_0, x_1, ...,$ los valores $f(x_0), f(x_1), ...,$ de la función f en estos nodos, y el punto $x^* \in [x_0, x_1]$.

Algoritmo 10

Poner $f(x_0) \to d_0$, $f(x_0) \to p$, $1 \to \psi$
Para $k = 0, 1, 2, ..,$ *hasta "terminar"* (k : *orden de la diferencia dividida*)
    poner $f(x_{k+1}) \to d_{k+1}$
    Para $j = k, k-1, .., 0$ (j : *nodo inicial*)
        calcular $\frac{d_{j+1}-d_j}{x_{k+1}-x_j} \to d_j$
poner $\psi.(x^* - x_k) \to \psi$
poner $p + d_0.\psi \to p : p_{k+1}(x^*)$

Criterios de parada

La expresión "hasta terminar"en el ciclo en k del algoritmo lleva implícito el cumplimiento de uno de los tres criterios de parada mencionados:

1) $|p_k(x^*) - p_{k-1}(x^*)| < \varepsilon$, si $|p_k(x^*) - p_{k-1}(x^*)|$ decrece

2) que se acaben los nodos ($k = n$)

3) que $|p_k(x^*) - p_{k-1}(x^*)| > |p_{k-1}(x^*) - p_{k-2}(x^*)|$, es decir, no decrezca.

En el ejemplo de aplicación de la fórmula de Newton, como $\overline{x} = 1 \in [0, 2]$ tomamos $x_0 = 0$ y calculamos

$p_0(1) = 1, p_1(1) = 2, p_2(1) = 2{,}67$ y $p_3(1) = 3{,}27$

Según el algoritmo anterior, es necesario calcular paralelamente la sucesión de las diferencias $|p_k(\overline{x}) - p_{k-1}(\overline{x})|$

$|p_1(1) - p_0(1)| = 1$

$|p_2(1) - p_1(1)| = 0{,}67 < 1$

$|p_3(1) - p_2(1)| = 0{,}60 < 0{,}67$

y se ve que son decrecientes y además se acabaron los nodos, luego $p_3(1) = 3{,}27$ es el mejor valor interpolado y tomamos entonces $f(1) \approx 3{,}27$.

¿Cuál es la precisión de esta aproximación?. Lo analizaremos estudiando el error del polinomio de interpolación.

Teorema 11 (de Faber) *Para cualquier conjunto de nodos*

$$a \leq x_0 < x_1 < ... < x_n \leq b$$

existe una función continua $f(x)$ *tal que la sucesión* $\{P_n(x)\}$ *no converge uniformemente a* $f(x)$.

Teorema 12 *Dada* $f \in C[a, b]$ *arbitraria, existe* $\{x_i\}_{i=0}^n$ *tal que* $\{P_n(x)\} \to f(x)$ *uniformemente.*

Observación 13 *El Teorema 11 indica que no hay esquemas de interpolación universales efectivos que tengan como base sólo valores de* f. *Ejemplo: la función de Runge.*

Observación 14 *Aunque el Teorema 12 representa un resultado más positivo, generalmente se desconoce cuál es el* x_i *que lo garantiza.*

La construcción práctica de $p_n(x)$ para n grande es un proceso tedioso. Por ello, los polinomios de interpolación de grado alto son de utilidad limitada en Análisis Numérico. Resulta más conveniente dividir el intervalo en subintervalos menores y usar polinomios de grado relativamente menor en cada subintervalo. De ahí la idea de la interpolación por tramos y los splines.

Fórmula de Newton para nodos equidistantes

La interpolación polinómica surgió de la necesidad de evaluar, en puntos intermedios, funciones que se conocían en forma tabular. Cuando los valores f_i corresponden a evaluaciones (mediciones) en una sucesión de valores equidistantes $x_i = x_0 + ih \quad (0 \leq i \leq n)$ de la variable independiente, se pueden hacer ciertas simplificaciones en la determinación del polinomio de interpolación. El espaciamiento h entre dos valores consecutivos x_i y x_{i+1} de la variable independiente se denomina paso de la tabla.

Se introduce el cambio de variable

$$s = s(x) = \frac{x - x_0}{h} \tag{1.30}$$

de donde

$$x = x(s) = x_0 + sh \tag{1.31}$$

Note que en (1.31), si $s \in \mathbb{N}$ entonces $x = x_s$ es nodo de interpolación. De acuerdo con (1.31), se tiene que

$$f(x_s) = f(x_0 + sh) \tag{1.32}$$

Como el cambio de variable (1.31) es lineal, convierte los polinomios de grado n en x, en polinomios de grado n en s.

Para determinar y evaluar el polinomio de grado menor o igual que n que interpola a f en los nodos equidistantes $x_0, ..., x_n$, no es necesario construir entonces una tabla de diferencias divididas, pues los denominadores $x_{j+k} - x_j$ se convierten en múltiplos de h:

$$x_{j+k} - x_j = kh$$

y basta en este caso construir una tabla de diferencias finitas.

Diferencias finitas

Las diferencias $f_{i+1} - f_i$ se denominan diferencias de primer orden. El valor $f_{i+1} - f_i$ se puede denotar de dos maneras

$$f_{i+1} - f_i = f(x_i + h) - f(x_i) = \begin{cases} \Delta f_i : \text{diferencia finita hacia adelante en el nodo } x_i \\ \nabla f_{i+1} : \text{ diferencia finita hacia atrás en el nodo } x_{i+1} \end{cases} \tag{1.33}$$

Las diferencias finitas de orden superior se forman con la ayuda de las relaciones de recurrencia siguientes

$$\Delta^n f_i = \Delta\left(\Delta^{n-1} f_i\right) = \Delta^{n-1} f_{i+1} - \Delta^{n-1} f_i$$
$$\nabla^n f_i = \nabla\left(\nabla^{n-1} f_i\right) = \nabla^{n-1} f_i - \nabla^{n-1} f_{i-1}$$

es decir, la $n-$ésima diferencia finita en x_i es igual a la primera diferencia finita de la diferencia finita de orden $n-1$ en x_i.

También para las diferencias finitas de orden cero se tiene que:

$$\Delta^0 f_i = f_i$$
$$\nabla^0 f_i = f_i$$

Luego, en general, para las diferencias finitas hacia adelante se tiene:

$$\Delta^n f_i = \begin{cases} f_i, \text{ para } n = 0 \\ \Delta\left(\Delta^{n-1} f_i\right) = \Delta^{n-1} f_{i+1} - \Delta^{n-1} f_i, \quad \text{para } n > 0 \end{cases} \tag{1.34}$$

y, análogamente, para las diferencias finitas hacia atrás:

$$\nabla^n f_i = \begin{cases} f_i, \text{ para } n = 0 \\ \nabla\left(\nabla^{n-1} f_i\right) = \nabla^{n-1} f_i - \nabla^{n-1} f_{i-1}, \quad \text{para } n > 0 \end{cases} \tag{1.35}$$

Tabla de diferencias finitas

x_i	f_i	$\Delta f_i = \nabla f_{i+1}$	$\Delta^2 f_i = \nabla^2 f_{i+2}$	$\ldots$
x_0	f_0			
x_1	f_1	$\Delta f_0 = \nabla f_1$		
x_2	f_2	$\Delta f_1 = \nabla f_2$	$\Delta^2 f_0 = \nabla^2 f_2$	
$\vdots$	$\vdots$	$\vdots$		
x_n	f_n	$\vdots$		

Para el cálculo y almacenamiento de la tabla de diferencias finitas en forma eficiente, es válido el mismo algoritmo estudiado para las diferencias divididas.

Entre las diferencias finitas y las diferencias divididas existe la siguiente relación:

$$f\left[x_i, \ldots, x_{i+k}\right] = \frac{1}{k! h^k} \Delta^k f_i \quad , \quad \forall k \geq 0 \tag{1.36}$$

donde k es el orden de la diferencia, i es el primer nodo, h es el paso de la tabla.

Esta relación se puede demostrar por inducción teniendo en cuenta la expresión recurrente que define las diferencias divididas de orden superior.

Resumiendo la relación entre las diferencias divididas, derivadas y diferencias finitas con respecto a los nodos $x_0, ..., x_k$ es :

$$f\left[x_0, ..., x_k\right] = \frac{f^{(k)}(\xi)}{k!} = \frac{\Delta^k f_0}{k! h^k} \quad , \quad x_0 < \xi < x_k \tag{1.37}$$

Fórmula de Newton en diferencias finitas hacia adelante

El polinomio de interpolación basado en los nodos $x_0, ..., x_n$ según la función de Newton en diferencias divididas se expresa como:

$$p_n(x) = \sum_{i=0}^{n} f\left[x_0, .., x_i\right] \prod_{j=0}^{i-1} (x - x_j)$$

a) Sustituyendo la diferencia dividida $f\left[x_0, ..., x_i\right]$ en términos de la diferencia finita correspondiente dada por (1.36), obtenemos

$$p_n(x) = \sum_{i=0}^{n} \frac{1}{i! h^i} \Delta^i f_0 \prod_{j=0}^{i-1} (x - x_j) \tag{1.38}$$

b) En términos de s, tenemos que

$$x - x_j = (x_0 + sh) - (x_0 + jh) = h\,(s - j) \tag{1.39}$$

Sustituyendo (1.39) en (1.38)

$$p_n\,(x) = p_n\,(x_0 + sh) = \sum_{i=0}^{n} \frac{1}{i!h^i} \Delta^i f_0 \prod_{j=0}^{i-1} h\,(s - j) \tag{1.40}$$

$$= \sum_{i=0}^{n} \Delta^i f_0 \frac{s!}{i!\,(s-i)!} \tag{1.41}$$

c) Definamos para $y \in \mathbb{R}, i \in \mathbb{N}$ la función binomial generalizada $b\,(y)$:

$$b\,(y) = \left(\begin{array}{c} y \\ i \end{array} \right) = \left\{ \begin{array}{c} 1, \quad \text{si } i = 0 \\ \prod_{j=0}^{i-1} \left(\frac{y-j}{j+1} \right) = \left(\frac{y}{1} \right) \left(\frac{y-1}{2} \right) ... \left(\frac{y-i+1}{i} \right), \quad \text{si } i > 0 \end{array} \right. \tag{1.42}$$

La palabra binomial se justifica, pues, si $y \in \mathbb{N}$ entonces la expresión (1.42) coincide con la del coeficiente binomial:

$$\left(\begin{array}{c} y \\ i \end{array} \right) = \frac{y!}{i!\,(y-i)}$$

d) Sustituyendo (1.42) en (1.41), obtenemos

$$p_n\,(x_0 + sh) = \sum_{i=0}^{n} \Delta^i f_0 \left(\begin{array}{c} s \\ i \end{array} \right), \; s = \frac{x - x_0}{h} \tag{1.43}$$

que, en forma desarrollada, nos da

$$p_n\,(x_0 + sh) = f_0 \left(\begin{array}{c} s \\ 0 \end{array} \right) + \Delta f_0 \left(\begin{array}{c} s \\ 1 \end{array} \right) + \Delta^2 f_0 \left(\begin{array}{c} s \\ 2 \end{array} \right) + ... + \Delta^n f_0 \left(\begin{array}{c} s \\ n \end{array} \right)$$

y sustituyendo las funciones binomiales,

$$p_n\,(x_0 + sh) = f_0 + s\Delta f_0 + \frac{s\,(s-1)}{2!} \Delta^2 f_0 + ... + \frac{s\,(s-1)\,...\,(s-n+1)}{n!} \Delta^n f_0 \tag{1.44}$$

Las expresiones (1.43) y (1.44) reciben el nombre de **fórmula de Newton en diferencias finitas hacia adelante.**

Los coeficientes $\Delta^i f_0$ de (1.43) y (1.44) son los que encabezan las columnas de la tabla de diferencias finitas construida a partir de los nodos $x_0, ..., x_n$.

La fórmula de Newton en diferencias finitas, además de utilizarse para el cálculo de valores interpolados, tiene aplicación en el cálculo de ceros, la derivación e integración aproximada, entre otros.

De manera análoga, considerando los nodos $x_{i-n}, \ldots, x_n$, se obtiene la **fórmula de Newton en diferencias finitas retrógradas o hacia atrás**

$$p_n(x) = \sum_{i=0}^{n} f[x_{n-i}, \ldots, x_n] \prod_{j=0}^{i-1} (x - x_{n-j})$$

$$f[x_{n-i}, \ldots, x_n] = \frac{\nabla^i f_n}{i! h^i} = \frac{\Delta^i f_{n-i}}{i! h^i}$$

$x - x_{n-j} = (x_n + sh) - (x_n - jh) = h(s + j)$

$$p_n(x_n + sh) = \sum_{i=0}^{n} \nabla^i f_n (-1)^i \begin{pmatrix} -s \\ i \end{pmatrix},$$

$s = \frac{\overline{x} - x_n}{h}$, $\begin{pmatrix} -s \\ i \end{pmatrix} = \frac{(-s)!}{i!(-s-i)!} = \begin{cases} 1, & \text{si } i = 0 \\ \left(\frac{-s}{1}\right)\left(\frac{-s-1}{2}\right) \ldots \left(\frac{-s-i+1}{i}\right), & \text{si } i > 0 \end{cases}$

$$p_n(x_n + sh) = f_n + s\nabla f_n + \frac{s(s+1)}{2}\nabla^2 f_n + \ldots + \frac{s(s+1)\ldots(s+n-1)}{n!}\nabla^n f_n$$

1.2. La forma de Hermite del polinomio de interpolación

Los polinomios de Hermite, la forma normal de Hermite y el spline cúbico de Hermite son llamados así en honor al matemático francés Charles Hermite (1822-1901), al cual deben su origen; quien además realizó investigaciones en temas como teoría de números, formas cuadráticas, polinomios ortogonales, álgebra, etc. En esta sección se presentan las nociones básicas de la interpolación de Hermite y análisis de su error, así como algunas de sus ventajas y desventajas. Para construir el polinomio de interpolación en la forma de Lagrange y en la forma de Newton, los datos son pares ordenados de valores de una función en un conjunto determinado de puntos. ¿Qué pasa si además se tienen valores de las derivadas? ¿Cómo se podría aprovechar esta nueva información?. La respuesta está en la construcción del polinomio de interpolación de Hermite, donde se aprovecha tal información.

Sean dados $m+1$ puntos y en cada punto $x_i, i = 0, .., m$ se conocen las derivadas hasta el orden $n_i - 1$, que se ordenan como datos de la forma:

$$(x_i, y_i^{(k)}),\ k = 0, \ldots, n_i - 1,\ i = 0, \ldots, m$$

Para $n + 1 := \sum_{i=0}^{m} n_i$ se construye el polinomio de interpolación de Hermite de orden menor o igual que n

$$P(x) = \sum_{i=0}^{m} \sum_{k=0}^{n_i - 1} y_i^{(k)} L_{ik}(x),$$

que satisfaga las siguientes condiciones de interpolación,

$$P^k(x_i) = y_i^{(k)},\ k = 0, \ldots, n_i - 1,\ i = 0, \ldots, m \tag{1.45}$$

Para cada nodo i se calculan n_i bases de Lagrange L_{ik}, para las cuales se cumple que:

$$L_{ik}^{(\sigma)}(x_j) = \begin{cases} 1 & i = j, \sigma = k \\ 0 & \text{si no} \end{cases}$$

Las condiciones (1.45) nos dan exactamente $\sum_{i=0}^{m} n_i = n + 1$ condiciones para los $n + 1$ coeficientes de $P(x)$, de manera que se espera la unicidad en la solución del problema.

$L_{ik}(x) \in \Pi_n$ es un polinomio de Lagrange generalizado que se construye mediante polinomios auxiliares

$$l_{ik}(x) := \frac{(x - x_i)^k}{k!} \prod_{\substack{j=0 \\ j \neq i}}^{m} \frac{(x - x_j)^{n_j}}{(x_i - x_j)^{n_j}},\ 0 \leq i \leq m, 0 \leq k \leq n_i - 1$$

Se define $L_{i,n_i-1}(x) := l_{i,n_i-1}(x)$, $i = 0, \ldots, m$, y de forma recursiva para $k = n_i - 2, n_i - 3, \ldots, 0$:

$$L_{ik}(x) := l_{ik}(x) - \sum_{\gamma=k+1}^{n_i-1} l_{ik}^{(\gamma)}(x_i) L_{i\gamma}(x).$$

Como se observa en la fórmula anterior, la sumatoria siempre comienza con la derivada de mayor orden que se conoce para el nodo i. En la literatura lo más frecuente es tener como datos los pares de números reales $(x_i, f_i^{(k)}), k = 0, 1; i = 0, .., m$.

Teorema 15 *Sea S un conjunto discreto de datos, para puntos distintos*

$$S = \{(x_i, f_i, f_i'),\ i = 0, \ldots, m\}$$

Entonces existe un único polinomio $p_{2m+1}(x)$ de grado menor o igual que $2m+1$ que interpola dichos datos, es decir

$$p_{2m+1}(x_i) = f(x_i) = f_i,\ p'_{2m+1}(x_i) = f'(x_i) = f_i',\ i = 0, \ldots m$$

En este caso el polinomio de Hermite se puede escribir como:

$$p_{2m+1}(x) = \sum_{i=0}^{m} [f(x_i)u_i(x) + f^{'}(x_i)v_i(x)] \tag{1.46}$$

con $u_i(x) = [1 - 2L_i'(x_i)(x - x_i)](L_i(x))^2$, $v_i(x) = (x - x_i)(L_i(x))^2$, $L_i(x) = \prod_{j \neq i} \left(\frac{x - x_j}{x_i - x_j}\right)$, $L_i \in \mathcal{P}_m$.

Existen tres formas de construir el polinomio de interpolación de Hermite

- Método de los coeficientes indeterminados (requiere resolver un Sistema de Ecuaciones Lineales.)
- Modificación de las bases de Lagrange (computacionalmente más engorroso)
- Modificación de las bases de Newton en diferencias divididas

A continuación se verá cómo se transforma la base de Newton en diferencias divididas para construir el polinomio de Hermite en el caso de $n_i = 2$ y $n+1$ nodos de interpolación. Sea la base de Newton

$$\{1, (x-x_0), (x-x_0)(x-x_1), \ldots, (x-x_0)\ldots(x-x_{n-1})\}$$

Denotando cada elemento de la base por $D_0(x) = 1, D_1(x) = (x-x_0), \ldots, D_n(x) = (x-x_0)\ldots(x-x_{n-1})$. La modificación sería:

$$\begin{aligned}
HD_0(x) &= 1 \\
HD_1(x) &= (x-x_0) \\
HD_2(x) &= (x-x_0)^2 \\
HD_3(x) &= (x-x_0)^2(x-x_1) \\
HD_4(x) &= (x-x_0)^2(x-x_1)^2 \\
HD_5(x) &= (x-x_0)^2(x-x_1)^2(x-x_2) \\
&\vdots \\
HD_{2n}(x) &= (x-x_0)^2(x-x_1)^2\ldots(x-x_{n-1})^2 \\
HD_{2n+1}(x) &= (x-x_0)^2(x-x_1)^2\ldots(x-x_{n-1})^2(x-x_n)
\end{aligned}$$

Para calcular los coeficientes hay que modificar el proceso de construcción de la tabla de diferencias divididas

x	DD orden 0	DD 1er	DD 2do	$\ldots$
x_0	$f[x_0]$			
		$f'(x_0)$		
x_0	$f[x_0]$			
		$f[x_0,x_1]$	$f[x_0,x_0,x_1]$	
x_1	$f[x_1]$			
		$f'(x_1)$	$f[x_0,x_1,x_1]$	
x_1	$f[x_1]$			
		$f[x_1,x_2]$	$f[x_1,x_1,x_2]$	
x_2	$f[x_2]$			
		$f'(x_2)$	$f[x_1,x_2,x_2]$	
x_2	$f[x_2]$			
		$f[x_2,x_3]$		
x_2	$f[x_3]$			
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

La columna de la diferencia dividida (DD) de segundo orden se calcula en la forma estándar, a partir de la primera, esto es

$$f[x_0,x_0,x_1] = \frac{f[x_0,x_1] - f'(x_0)}{x_1 - x_0}$$

$$f[x_0,x_1,x_1] = \frac{f'(x_1) - f[x_0,x_1]}{x_1 - x_0}$$

. Entonces los coeficientes del polinomio de Hermite serán los que aparecen en la diagonal de la tabla como es usual

$$p_{2n+1}(x) = f[x_0] + f'(x_0)(x - x_0) + \frac{f[x_0, x_1] - f'(x_0)}{x_1 - x_0}(x - x_0)^2 + \dots$$

Ejemplo 16 *Sea* $p(0) = -1, p(1) = 0$, $p'(1) = \alpha$, $\alpha \in \mathbb{R}$. *¿Cuál es el grado del polinomio de grado mínimo de Hermite?*

El polinomio de Hermite cumple que $n + 1 := \sum_{i=0}^{m} n_i$, *donde* n *es el grado del polinomio de Hermite. Entonces en nuestro ejemplo:*

$i = 0$, $n_0 = 1$, pues $k = 0$, es decir, en el punto x_0 se conoce el valor de $f(x_0)$.
$i = 1$, $n_1 = 2$, pues $k = 0, 1$, es decir, en el punto x_1, se conoce el valor de $f(x_1)$ y $f'(x_1)$.

Luego, sustituyendo en la condición se tiene que: $\sum_{i=0}^{1} n_i = 3$ y por tanto el grado del polinomio de Hermite es 2.

Comentarios

El uso de la fórmula de Lagrange para calcular valores interpolados con una precisión prefijada no es eficiente, ya que, si es necesario aumentar el grado del polinomio de interpolación para alcanzar la precisión deseada, no se pueden aprovechar los cálculos realizados para el polinomio de menor grado. La fórmula de Newton resuelve este problema mediante el algoritmo de interpolación con número creciente de nodos y el uso de una tabla de diferencias divididas.

Sin embargo, la construcción práctica de $p_n(x)$ para n grande, no solo es un proceso tedioso, sino que presenta algunas limitaciones:

- para nodos de interpolación impuestos y uniformemente espaciados pueden aparecer fuertes oscilaciones.
- pequeñas perturbaciones en los datos pueden generar fuertes variaciones del polinomio interpolante.
- es difícil que la interpolación preserve la forma local de los datos (positividad, monotonía, convexidad), es decir, en cada subintervalo definido por los nodos de interpolación.

Por ello los polinomios de interpolación de grado alto son de utilidad limitada en Análisis Numérico. Resulta más conveniente dividir el intervalo en subintervalos menores y usar polinomios de grado relativamente menor en cada subintervalo. De ahí la idea de la interpolación por tramos y los splines.

1.3. Estabilidad de la interpolación polinomial

La estabilidad de un problema (para datos dados), nos habla del impacto que causan pequeñas perturbaciones de los datos en el resultado final. En el contexto de la interpolación polinomial nos dirá hasta que punto perturbaciones en los datos afectan los valores de la función de interpolación en otros puntos. Para estudiar la sensibilidad de la interpolación polinomial en términos cuantitativos se necesitan las normas en el espacio vectorial de las funciones continuas $C(I), I \subset \mathbb{R}$

1.4. Interpolación por tramos

En cualquier representación que se utilice para construir el polinomio de interpolación: la fórmula de Lagrange, la fórmula de Newton en sus diferentes variantes o la fórmula de Hermite, interesa poder estimar el error de valores interpolados $p_n(x^*)$ en puntos x^* que no sean nodos de interpolación, en otras palabras conocer cuán bien aproxima el polinomio de interpolación a la función dada. Como se observa en la Figura (1.8), cuando se tienen muchos nodos y equidistantes, el polinomio de interpolación de alto orden, experimenta fuertes oscilaciones, no brinda una buena aproximación. Una solución a este problema es construir polinomios de menor orden, es decir, polinomios que interpolen por tramos.

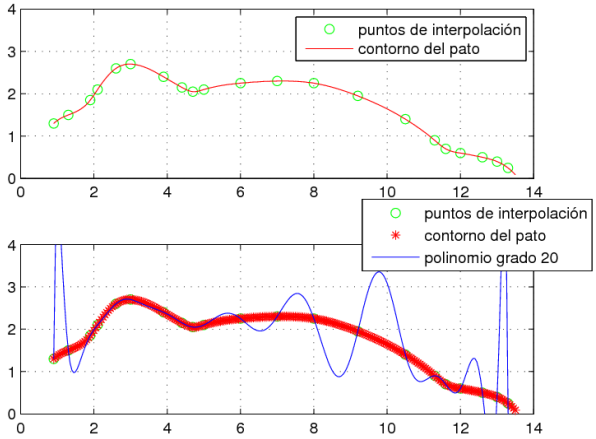


Figura 1.8: Oscilaciones que se presentan al aproximar por polinomios de orden elevado

1.4.1. El error del polinomio de interpolación

Sea $f(x)$ una función real definida en el intervalo $I = [a, b]$ y sean $n + 1$ puntos distintos $x_0, x_1, \ldots, x_n$ de I. Sea $p_n(x)$ el polinomio de grado menor o igual que n que interpola a f en los nodos $x_0, \ldots, x_n$; el error $e_n(x)$ del polinomio de interpolación se define como:

$$e_n(x) := f(x) - p_n(x). \tag{1.47}$$

Consideremos ahora el punto x^*, que no es un nodo de interpolación. Si $p_{n+1}(x)$ es el polinomio de grado $\leq n + 1$ que interpola a f con nodos $x_o, \ldots, x_n, x^*$, entonces

$$p_{n+1}(x^*) = f(x^*), \tag{1.48}$$

y de acuerdo con la fórmula de Newton

$$p_{n+1}(x) = p_n(x) + f[x_0, \ldots, x_n, x^*](x - x_0) \cdots (x - x_n). \tag{1.49}$$

Luego, de (1.48) y (1.49) obtenemos

$$f(x^*) = p_{n+1}(x^*) = p_n(x^*) + f[x_0, \ldots, x_n, x^*](x^* - x_0) \ldots (x^* - x_n), \tag{1.50}$$

y sustituyendo (1.50) en (1.47),

$$e_n(x^*) = \{p_n(x^*) + f[x_0, \ldots, x_n, x^*](x^* - x_0) \cdots (x^* - x_n)\} - p_n(x^*)$$

es decir, $\forall x^* \neq x_0, \ldots, x_n$, se tiene que

$$e_n(x^*) = f[x_0, \ldots, x_n, x^*](x^* - x_0) \cdots (x^* - x_n). \tag{1.51}$$

Esta expresión muestra que el error en el punto x^* es *parecido al próximo término* de la fórmula de Newton, $f[x_0, \ldots, x_n, x_{n+1}](x^* - x_0) \cdots (x^* - x_n)$, es decir si se considera el polinomio de grado $n+1$, evaluado en x^*. Lo anterior justifica la forma de proceder en la interpolación con número creciente de nodos en aras de obtener una aproximación de $f(x^*)$ lo más precisa posible.

La expresión (1.51) no puede ser evaluada, a menos que se conozca el valor $f(x^*)$ y con él se calcule $f[x_0, \ldots, x_n, x^*]$. Pero como veremos a continuación, el número $f[x_0, \ldots, x_n, x^*]$ está intimamente relacionado con la derivada de orden $(n+1)$ de $f(x)$, y usando esta información, podemos a veces estimar $e_n(x^*)$.

Teorema 17 *Sea $f(x)$una función real, continua sobre un intervalo $[a, b]$ y $(n+1)$ veces diferenciable en (a, b). Si $p_n(x)$ es el polinomio de grado $\leq n$ que interpola a $f(x)$ en los $n+1$ puntos distintos $x_0, \ldots, x_n$ en $[a, b]$, entonces para todo $x^* \in [a, b]$, existe $\xi = \xi(x^*) \in (a, b)$ tal que*

$$e_n(x^*) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^{n} (x^* - x_j) \tag{1.52}$$

Demostración 18 *Sea $x^* \neq x_0, \ldots, x_n$ un punto en $[a, b]$. Definamos una función $g(t)$*

$$g(t) = [p_n(t) - f(t)] - \frac{w(t)}{w(x^*)} [p_n(x^*) - f(x^*)]$$

donde $w(x) = (x - x_0) \ldots (x - x_n)$, entonces $g(t)$ tiene $(n+2)$ ceros en los nodos $I = [x_0, \ldots, x_n, x^*]$. Aplicando el Teorema de Rolle, $g'(t)$ tiene por lo menos $(n+1)$ ceros en I , $g''(t)$ tiene por lo menos (n) ceros en I y así repetidamente obtenemos que $g^{(n+1)}(t)$ tiene al menos una raiz $\xi \in I$ y como la derivada de orden $(n+1)$ de $p_n(t)$, $p_n^{(n+1)}(t) \equiv 0$, se obtiene

$$\begin{aligned} g^{(n+1)}(\xi) &= -f^{(n+1)}(\xi) - \frac{(n+1)!}{w(x^*)} [p(x^*) - f(x^*)] \\ &= 0 \end{aligned}$$

Por tanto,

$$f(x^*) - p_n(x^*) = \frac{f^{(n+1)}(\xi)}{(n+1)!} w(x^*) \tag{1.53}$$

Teorema 19 *Sea $f(x)$una función que toma valores reales, continua sobre un intervalo $[a, b]$ y k veces diferenciable en (a, b). Si $x_0, \ldots, x_k$ son $k+1$ puntos distintos en $[a, b]$, entonces existe $\xi \in (a, b)$, tal que*

$$f[x_0, \ldots, x_k] = \frac{f^{(k)}(\xi)}{k!} \tag{1.54}$$

Demostración 20 *Si consideramos (1.50)*

$$f(x^*) = p_{n+1}(x^*) = p_n(x^*) + f[x_0, \ldots, x_n, x^*](x^* - x_0)\ldots(x^* - x_n)$$

tenemos que

$$f(x^*) - p_n(x^*) = f[x_0, \ldots, x_n, x^*](x^* - x_0)\ldots(x^* - x_n),$$

y comparando con (1.53), se tiene que

$$f[x_0, \ldots, x_n, x^*] = \frac{f^{(n+1)}(\xi)}{(n+1)!}, \text{ para un } \xi \in I = [x_0, \ldots, x_n, x^*]$$

con lo que se cumple, en general,

$$f[x_0, \ldots, x_n] = \frac{f^{(n)}(\xi)}{(n)!}, \text{ para un } \xi \in I = [x_0, \ldots, x_n]$$

Observación 21 *Observe que para $k = 1$ este es precisamente el teorema del valor medio para derivadas. Tomando $a = \min_i x_i, b = \max_i x_i$, se desprende que el punto desconocido ξ puede suponerse que está entre los nodos x_i.*

Es importante apuntar que $\xi = \xi(x^*)$ depende del punto x^* en el cual se requiere estimar el error. Esta dependencia ni siquiera tiene que ser continua. Además, ξ pertenece al menor intervalo que contiene a x^* y a los puntos de interpolación.

La expresión (1.52) es de utilidad práctica limitada pues, en general, $f^{(n+1)}(x)$ y el punto ξ se desconocen. Una cota superior de $e_n(x)$ está dada por

$$|e_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \max_{x\in[a,b]} |(x - x_0)\cdots(x - x_n)| \tag{1.55}$$

donde $M_{n+1} = \max_{x\in[a,b]} \left|f^{(n+1)}(x)\right|$, pero tampoco resulta fácil, en algunos casos, hallar M_{n+1}.

Ejemplo 22 *Hallar una cota para el error en la interpolación lineal.*

El polinomio de interpolación lineal $f(x)$ en x_0 y x_1 es

$$p_1(x) = f(x_0) + f[x_0, x_1](x - x_0)$$

La ecuación (1.52) nos da la fórmula

$$e_1(x^*) = \frac{f''(\xi)}{2!}(x^* - x_0)(x^* - x_1)$$

donde ξ depende de x^*, siendo x^* un punto entre x_0 y x_1. Si conocemos que $\left|f''(x)\right| \leq M$ en $[x_0, x_1]$, entonces

$$e_1(x^*) \leq \frac{M}{2}|(x^* - x_0)(x^* - x_1)|$$

El valor máximo de $|(x^* - x_0)(x^* - x_1)|$ para $x^* \in [x_0, x_1]$ se alcanza en $x^* = \frac{x_0+x_1}{2}$, y es igual a $\frac{(x_1-x_0)^2}{4}$. Se concluye que para cualquier $x^* \in [x_0, x_1]$

$$e_1(x^*) \leq \frac{M}{8}\left|(x_1 - x_0)^2\right|$$

Observación 23
- $f[x_0, \ldots, x_n, x^*]$ *puede estimarse mediante* $f[x_0, \ldots, x_n, x_{n+1}]$ *para* $x_{n+1} \approx x^*$ *(esto conlleva añadir un punto y una diagonal más en la tabla de diferencias divididas).*
- $|e_n(x^*)|$ *aumenta cuando* x^* *está lejos de los puntos de interpolación.*

¿Cómo reducir entonces el error de interpolación?
La expresión (1.52) del error en el punto x^* se puede separar en dos partes:

$$e_n(x^*) = \left(\frac{f^{(n+1)}(\xi)}{(n+1)!}\right)((x^* - x_0)\cdots(x^* - x_n))$$

donde el primer factor depende de f y no se puede alterar y el segundo depende sólo de los nodos. La reducción del error se puede llevar a cabo sólo variando el segundo factor, lo que se puede realizar por diferentes vías:

1. Escogiendo los nodos en orden de proximidad al punto x^* donde se quiere interpolar, lo que requerirá la reordenación de los nodos.

2. Escogiendo los nodos de modo que se minimice $\prod_{j=0}^{N}(x - x_j)$, lo que da lugar a los llamados polinomios de interpolación de Chebyshev (válido sólo si se dispone de la expresión analítica de f y de los ceros de los polinomios de Chebyshev en $[x_0, x_n]$).

3. Particionando el intervalo de interpolación en tramos, usando polinomios de grado bajo en cada subintervalo, con el objetivo de reducir el valor de la productoria, al ser menor la longitud de cada subintervalo y tener menos subintervalos. Esto da lugar a la interpolación por tramos.

1.4.2. Interpolación con spline cúbico

Cuando el grado del polinomio de interpolación es alto, el error en puntos intermedios entre los nodos puede llegar a ser muy grande (recordar la función de Runge). Mientras mayor sea n, más conflictiva es la presencia de máximos y mínimos en el polinomio de interpolación. En muchos casos, la naturaleza de los datos indica que no se justifica el uso de polinomios de grado alto; por ejemplo, el **problema cartográfico de aproximación de la línea costera** ya que la trayectoria que se describe es muy irregular como se observa en la figura (1.9).

Se presenta entonces la necesidad de un tipo de aproximación por interpolación con polinomios de grado pequeño sobre intervalos de longitud reducida.

Si tenemos una función f dada mediante una tabla de valores $(x_i, f_i), 0 \leq i \leq N$, en lugar de construir el polinomio de interpolación de grado N, buscamos un sistema de polinomios de grado k :

$$\left\{S_i^{(k)}(x)\right\}_{i=1}^{N} \quad , \quad k \ \text{ fijo},$$

tales que $S_i^{(k)}(x)$ coincida con f y con las derivadas de f hasta el orden $(k-1)$ en los puntos (x_{i-1}, f_{i-1}) y (x_i, f_i), y tal que el sistema completo sea continuo y diferenciable al menos $k-1$ veces en el intervalo $[x_0, x_N]$.

Es decir, estamos planteando la aproximación por tramos mediante polinomios sujeta a restricciones de interpolación y de suavidad.

Figura 1.9: Aproximación del contorno de la bahía de La Habana

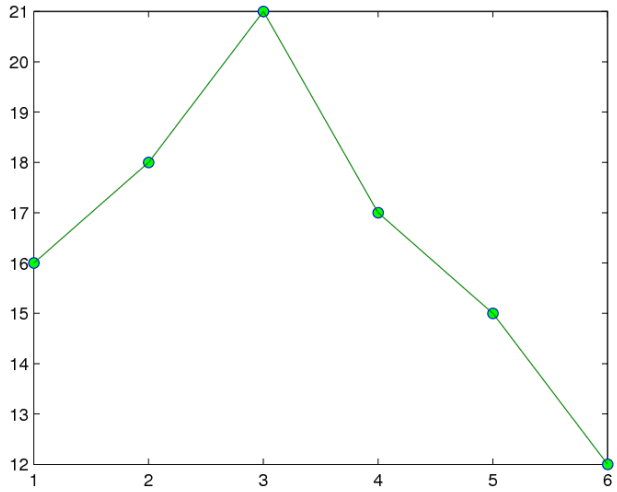


Figura 1.10: Interpolación lineal por tramos

Note que el polinomio particular $S_i(x)$ está asociado a un intervalo $[x_{i-1}, x_i]$ que, en general, será distinto al $S_j(x)$ asociado al intervalo $[x_{j-1}, x_j]$. Este tipo de interpolación es conocida como interpolación por spline, la cual es usada fundamentalmente con fines gráficos, es decir cuando se quiere unir un conjunto de puntos dados mediante curvas o superficies con un determinado orden de diferenciabilidad (lo suficientemente suaves).

Definición 24 *Un polinomio de grado k definido por tramos que tiene derivadas continuas hasta el orden $k-1$ es llamado spline de grado k.*

La elección más popular para k es 3, caso en el cual tratamos con un conjunto de polinomios de tercer grado definidos localmente, que posee continuidad global y primera y segunda derivadas continuas globalmente.

Definición 25 *Sea $f : [a,b] \to \mathbb{R}$ y $\Delta := \{a = x_0 < x_1 < ... < x_n = b\}$ una partición del intervalo $[a,b]$. Se llama spline cúbico S que interpola a f en los nodos de Δ a: $S : [a,b] \to \mathbb{R}$; $S \in C^2[a,b]$ si:*

- *Para cada* $i = 1, ..., n$, $S(x)$ *es un polinomio cúbico, denotado por* S_i *en* $[x_{i-1}, x_i)$.
- $S_i(x_i) = f_i$, $i = 0, ..., n$,
- $S_i(x_i) = S_{i+1}(x_i) = f_i$, $i =, 1, ..., n-1$,
- $S_i^{'}(x_i) = S_{i+1}^{'}(x_i)$, $i = 1, ..., n-1$,
- $S_i^{''}(x_i) = S_{i+1}^{''}(x_i)$, $i = 1, ..., n-1$.

Para completar las $4n$ condiciones que se necesitan para determinar los coeficientes de los polinomios cúbicos en los n subintervalos se deben adicionar dos condiciones en los extremos (apellidan al spline cúbico), que pueden ser entre otras

- $S^{''}(x_0) = S^{''}(x_n) = 0$, natural (frontera libre).
- $S'(x_0) = f'(x_0)$ y $S'(x_n) = f'(x_n)$ (frontera apoyada).
- $S^{(k)}(x_0) = S^{(k)}(x_n)$, para $k = 0, 1, 2$ (periódico). En este caso se presupone $f_0 = f_n$.

Isaac Schoenberg es conocido como el padre de los spline; en un artículo publicado en 1946, fue el primero en acuñar este término y reconocer la importancia de las funciones spline en el análisis matemático y en la teoría de aproximación, así como su uso en la solución numérica de ecuaciones diferenciales con condiciones iniciales y ó de fronteras. Schoenberg señaló que la función así definida y denominada era el equivalente matemático de un curvígrafo flexible usado hacía mucho tiempo por ingenieros y arquitectos; conocida en el lenguaje común como "la culebra".

Deducción del spline cúbico natural

Sea la función $f : \mathbb{R} \longrightarrow \mathbb{R}$ dada mediante la tabla

x	x_0	x_1	x_2	$\cdots$	x_N
$f(x)$	f_0	f_1	f_2	$\cdots$	f_N

Considerando el intervalo particular $[x_{i-1}, x_i]$, el spline cúbico correspondiente a ese intervalo será:

$$S_i(x) = \begin{cases} a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i, \text{ si } x \in [x_{i-1}, x_i] \\ 0, \text{ si } x \notin [x_{i-1}, x_i] \end{cases}, 1 \leq i \leq N \qquad (1.56)$$

Esta expresión contiene cuatro coeficientes desconocidos en el tramo i-ésimo, luego en total son $4N$ coeficientes a calcular. Para ello contamos con dos condiciones que se derivan de la interpolación:

$$S_i(x_{i-1}) = f_{i-1} \quad \text{y} \quad S_i(x_i) = f_i, \qquad (1.57)$$

que suman $2(N-1)$ en los nodos interiores más dos nodos extremos, en total $2N$. Las dos condiciones faltantes se obtienen eligiendo los coeficientes a_i, b_i, c_i, d_i, de modo que la primera y segunda derivada de $S(x)$ en el nodo x_i, coincidan en los tramos contiguos i e $i+1$, para lograr suavidad:

$$S_i{}'(x_i) = S_{i+1}'(x_i) \quad \text{y} \quad S\,{}''_i(x_i) = S\,{}''_{i+1}(x_i), \qquad (1.58)$$

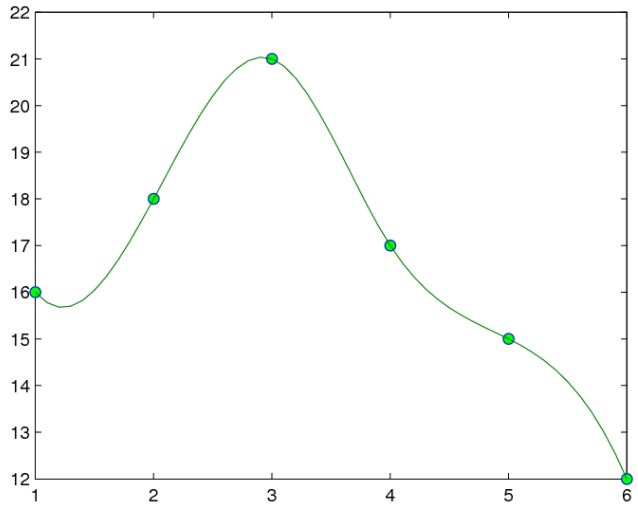


Figura 1.11: Aproximación por tramos con Spline Cúbico

en total, $2(N-1)$ pues son $N-1$ nodos interiores. Luego, se tienen en total $4(N-1)+2 = 4N-2$ condiciones, y faltan dos para que los $4N$ coeficientes puedan ser determinados unívocamente. Sobre esto volveremos posteriormente. Derivando la expresión (1.56) en el intervalo $[x_{i-1}, x_i]$

$$S_i'(x) = 3a_i(x-x_i)^2 + 2b_i(x-x_i) + c_i \tag{1.59}$$

$$S\,''_i(x) = 6a_i(x-x_i) + 2b_i \tag{1.60}$$

evaluando $S_i(x)$ en $x = x_i$, $S_i(x_i) = d_i$, y teniendo en cuenta (1.57), se tiene

$$d_i = f_i. \tag{1.61}$$

Evaluando $S\,''_i(x)$ en x_{i-1} y x_i, y denotando

$$\begin{aligned} t_{i-1} &= S_i''(x_{i-1}) = 6a_i(x_{i-1}-x_i) + 2b_i \\ t_i &= S_i''(x_i) = 2b_i \end{aligned}$$

de donde,

$$b_i = \frac{t_i}{2} \quad \text{y} \quad a_i = \frac{t_i - t_{i-1}}{6(x_i - x_{i-1})}. \tag{1.62}$$

Evaluando ahora $S_i(x)$ en x_{i-1}

$$S_i(x_{i-1}) = a_i(x_{i-1}-x_i)^3 + b_i(x_{i-1}-x_i)^2 + c_i(x_{i-1}-x_i) + d_i$$

y denotando $h_i = x_i - x_{i-1}$, se obtiene

$$S_i(x_{i-1}) = -a_i h_i^3 + b_i h_i^2 - c_i h_i + f_i$$

De (1.57), se tiene que $S_i(x_{i-1}) = f_{i-1}$ y despejando c_i,

$$c_i = \frac{f_i - f_{i-1}}{h_i} + \left(\frac{2t_i + t_{i-1}}{6}\right) h_i. \tag{1.63}$$

Las expresiones (1.61), (1.62) y (1.63) nos dan los cuatro coeficientes del tramo i-ésimo del spline en términos de las variables auxiliares t_{i-1} y t_i.

Para determinar ahora estas variables auxiliares, utilicemos las exigencias de suavidad en los nodos internos. La primera derivada del spline deberá cumplir (1.58) en $x_i, (1 \leq i \leq N)$:

$$\begin{aligned} S_i'(x_i) &= c_i \\ S_{i+1}'(x_i) &= 3a_{i+1}h_{i+1}^2 - 2b_{i+1}h_{i+1} + c_{i+1}, \end{aligned}$$

$$c_i = 3a_{i+1}h_{i+1}^2 - 2b_{i+1}h_{i+1} + c_{i+1},$$

y sustituyendo los coeficientes a, b, c en términos de los t en esta expresión se obtiene

$$\begin{aligned} & \frac{f_i - f_{i-1}}{h_i} + \left(\frac{2t_i + t_{i-1}}{6}\right) h_i \\ = \; & 3\left(\frac{t_{i+1} - t_i}{6h_{i+1}}\right) h_{i+1}^2 - 2\left(\frac{t_{i+1}}{2}\right) h_{i+1} \\ & + \frac{f_{i+1} - f_i}{h_{i+1}} + \left(\frac{2t_{i+1} + t_i}{6}\right) h_{i+1}, \end{aligned}$$

donde al escribir las expresiones de a_{i+1}, b_{i+1} y c_{i+1} está implícita la exigencia (1.58) para la segunda derivada, ya que $S\,''_i(x_i) = t_i = S\,''_{i+1}(x_i)$.

Agrupando en las variables t_{i-1}, t_i y t_{i+1}, se obtiene

$$\begin{aligned} & h_i t_{i-1} + 2(h_i + h_{i+1})t_i + h_{i+1}t_{i+1} \\ = \; & 6\left(\frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i}\right), \; (1 \leq i \leq N-1). \end{aligned} \tag{1.64}$$

Esta expresión representa un sistema de $N-1$ ecuaciones lineales en las $N+1$ incógnitas $t_o, t_1, \ldots, t_{N-1}, t_{N.}$, por lo que hacen falta 2 condiciones adicionales para que la solución sea única. Si tomamos

$$t_0 = S_0''(x_0) = t_N = S_N''(x_N) = 0, \tag{1.65}$$

o sea, curvatura nula en los nodos extremos, se obtiene el llamado ***spline cúbico natural***.

$$f[x_{i-1}, x_i] = \frac{f_i - f_{i-1}}{x_i - x_{i-1}}$$

El sistema (1.64)-(1.65) representado matricialmente será:

$$\begin{bmatrix} 2(h_1+h_2) & h_2 & 0 & \cdots & \cdots \\ h_2 & 2(h_2+h_3) & h_3 & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & 0 & h_{N-2} & 2(h_{N-2}+h_{N-1}) & h_{N-1} \\ \cdots & \cdots & 0 & h_{N-1} & 2(h_{N-1}+h_N) \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_{N-2} \\ t_{N-1} \end{bmatrix} =$$

$$= 6 \begin{bmatrix} (f[x_1, x_2] - f[x_o, x_1]) \\ (f[x_2, x_3] - f[x_1, x_2]) \\ \cdots \\ (f[x_{N-2}, x_{N-1}] - f[x_{N-3}, x_{N-2}]) \\ (f[x_{N-1}, x_N] - f[x_{N-2}, x_{N-1}]) \end{bmatrix}. \tag{1.66}$$

donde $f[x_{i-1}, x_i] = \frac{f_i - f_{i-1}}{x_i - x_{i-1}}$. La matriz del sistema (1.66) es definida positiva, lo que garantiza solución única. Dicha matriz es simétrica y tridiagonal, por lo que resulta adecuada la aplicación del método de Cholesky o del método de Gauss adaptado para matrices tridiagonales. También es aplicable la iteración de Gauss-Seidel, al ser la matriz de diagonal estrictamente dominante por filas $(2(h_i + h_{i+1}) > h_i + h_{i+1})$.

Resolviendo el sistema, se obtienen $t_1, t_2, \ldots, t_{N-1}$, y sustituyendo estos valores, además de $t_0 = t_N = 0$, en las expresiones (1.61), (1.62) y (1.63) que definen a_i, b_i, c_i, d_i, quedan determinados los polinomios cúbicos $S_i(x), 1 \leq i \leq N$, y con ello el spline $S(x)$. Los coeficientes a, b, c, d de cada tramo se almacenan en memoria con vista al cálculo de valores interpolados $S(x^*)$ para $x^* \in [x_{i-1}, x_i]$.

Ejemplo 26 *Hallar el spline cúbico natural de interpolación de la función dada por la siguiente tabla:*

x	*25*	*36*	*49*	*64*	*81*
f(x)	*5*	*6*	*7*	*8*	*9*

Para plantear el sistema lineal a resolver, es necesario calcular las primeras diferencias divididas:

i	x_i	$f(x_i)$	$f[,]$	h_i
0	25	5	–	–
1	36	6	1/11	11
2	49	7	1/13	13
3	64	8	1/15	15
4	81	9	1/17	17

El sistema lineal será

$$\begin{bmatrix} 2(11+13) & 13 & 0 \\ 13 & 2(13+15) & 15 \\ 0 & 15 & 2(15+17) \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = 6 \begin{bmatrix} 1/13 - 1/11 \\ 1/15 - 1/13 \\ 1/17 - 1/15 \end{bmatrix},$$

resolviéndolo se obtiene: $t_1 = -,001595,\ t_2 = -,000567,\ t_3 = -,000603$,teniendo en cuenta además que $t_o = t_4 = 0$, y sustituyendo en las expresiones para a_i, b_i, c_i, d_i, se obtienen los coeficientes del spline en cada tramo:

i	*intervalo*	a_i	b_i	c_i	d_i
1	$[25, 36]$	$2{,}417 \times 10^{-5}$	$-,000798$	,08506	6
2	$[36, 49]$	$-1{,}318 \times 10^{-5}$	$-,000284$	,07099	7
3	$[49, 64]$	$-4{,}073 \times 10^{-7}$	$-,000302$	,06227	8
4	$[64, 81]$	$-5{,}913 \times 10^{-6}$	0	,05709	9

Entonces, si se desea interpolar en $x^* = 55$, habrá que evaluar en el tramo $[49, 64]$ que lo contiene, es decir,

$$\begin{aligned} S(55) &= S_3(55) = a_3(55 - x_3)^3 + b_3(55 - x_3)^2 + c_3(55 - x_3) + d_3 \\ &= -4{,}073 \times 10^{-7}(-9)^3 - {,}000302(-9)^2 + {,}06227(-9) + 8 \\ &= 7{,}4155. \end{aligned}$$

Observando que la función tabulada no es más que $f(x) = \sqrt{x}$, el valor obtenido es una buena aproximación del valor exacto de $\sqrt{55} = 7{,}416$.

1.4.3. Interpolación cúbica de Hermite por tramos

Denotando $h_k = x_{k+1} - x_k$ y $d_k = P'(x_k)$, podemos escribir un polinomio cúbico en las variables $s = x - x_k$ y $h = h_k$ definido en el intervalo $x_k \leq x \leq x_{k+1}$,

$$\begin{aligned} P(x) &= \frac{3hs^2 - 2s^3}{h^3} y_{k+1} + \frac{{}^{h3} - 3hs^2 + 2s^3}{h^3} y_k \\ &\quad + \frac{s^2(s-h)}{h^2} d_{k+1} + \frac{s(s-h)^2}{h^2} d_k \end{aligned}$$

$P(x)$ es un polinomio cúbico en s y por tanto en x, que satisface cuatro condiciones de interpolación

$$\begin{aligned} P(x_k) &= y_k,\ P(x_{k+1}) = y_{k+1}, \\ P^{'}(x_k) &= d_k,\ P^{'}(x_{k+1}) = y_{k+1} \end{aligned} \tag{1.67}$$

Como se había visto anteriormente los polinomios de interpolación que satisfacen condiciones de interpolación para las derivadas, se conocen como polinomios de interpolación de Hermite, por tanto si se conocen las condiciones (1.67), entonces estaríamos en presencia de un polinomio cúbico de interpolación por tramos de Hermite. Si los valores de la derivada no son dados, es necesario definir la pendiente de alguna forma; en (NC with MatLab, Cleve Moler 2004) se presenta una de las formas de hacerlo.

Se define $\delta_k = \frac{y_{k+1} - y_k}{h_k}$, es decir la diferencia dividida de primer orden, y la idea clave es determinar la pendiente d_k de manera que los valores de la función no sobrepasen los valores de los datos, por lo menos localmente. Si δ_k y δ_{k-1} tienen signos opuestos o uno de los dos es cero, entonces x_k es un punto de mínimo o máximo local, y hacemos

$$d_k = 0.$$

Si δ_k y δ_{k-1} tienen el mismo signo y los dos subintervalos tienen el mismo largo entonces

$$d_k = \frac{1}{2}\left(\frac{1}{\delta_{k-1}} + \frac{1}{\delta_k}\right)$$

es decir se toma como la media armónica entre los dos valores discretos de las pendientes. Si δ_k y δ_{k-1} tienen el mismo signo y los dos subintervalos tienen tamanos diferentes entonces

$$\frac{w_1 + w_2}{d_k} = \frac{w_1}{\delta_{k-1}} + \frac{w_2}{\delta_k}$$

donde $w_1 = 2h_k + h_{k-1}$, $w_2 = h_k + 2h_{k-1}$. Es decir d_k es una media armónica pesada.

1.5. Ejercicios para el estudio independiente

1. Verifique que los polinomios $p(x) = 5x^3 - 27x^2 + 45x - 21$ y $q(x) = x^4 - 5x^3 + 8x^2 - 5x + 3$ satisfacen la condición de interpolación para los siguientes datos:

x	1	2	3	4
$f(x)$	2	1	6	47

   a) Explique por qué no se viola la unicidad del polinomio de interpolación.

   Orientación: Utilice la función **polyval** de Matlab.

2. Dado un conjunto de puntos de la forma $\{(x_i, y_i), i = \overline{1, n}\}$,

   a) Escriba un programa en MATLAB que, usando las funciones `vander` y la división izquierda \, calcule los coeficientes del polinomio de interpolación que interpole los puntos.

   b) Escriba un programa en MATLAB que devuelva la expresión analítica del polinomio de interpolación de Lagrange, que interpole los puntos.

   c) Utilice los programas creados por usted para obtener polinomios de interpolación para los siguientes datos:

x	0	1	2	3	4	5	6	7	8	9	10
y	1	2	3	4	5	6	7	8	9	10	11

   d) Note que el polinomio que resulta de usar el algoritmo implementado en el inciso $2a$) difere del que se obtiene si se usa el resultado del inciso $2b$). ¿Por qué ocurre esto?

3. Se desea medir el comportamiento de una motocicleta de carrera, para lo cual se decidió observar la velocidad y la distancia recorrida cada cierto tiempo, obteniéndose así la siguiente tabla.

t (s)	0	3	5	8	13
s (m)	0	225	383	623	993
v (m/s)	75	77	80	74	72

   a) Utilice la interpolación de Hermite para predecir cuánto había recorrido la motocicleta después de 10s y cuál era su velocidad en ese momento.

   b) ¿ Cómo valora los resultados obtenidos? ¿Serán una buena aproximación?

4. Se desea conocer el valor de $f(2{,}5)$, pero solo se dispone de la siguiente tabla de valores:

x	1	2	3	4
f(x)	12	15	25	60

   a) Si tuviera que estimar el valor de $f(2{,}5)$ utilizando un solo nodo de interpolación, ¿cuál escogería y por qué? Estime el valor de $f(2{,}5)$ utilizando un nodo de interpolación.

b) Si tuviera que estimar el valor de $f(2{,}5)$ utilizando solamente dos nodos de interpolación, ¿cuáles escogería y por qué? Estime el valor de $f(2{,}5)$ utilizando dos nodos de interpolación.

c) A partir de los dos incisos anteriores, deduzca cuáles serán los dos incisos siguientes, tomando en cuenta que solo se dispone de 4 nodos de interpolación.

d) Responda los dos incisos deducidos por usted en el inciso anterior.

e) De las estimaciones realizadas por usted en los incisos anteriores, ¿cuál cree que tiene una mayor precisión? ¿Por qué?

Bibliografía recomendada

- *Elementary Numerical Analysis, An algorithmic approach. 3rd Edition.* S. D. Conte y Carl de Boor. McGraw-Hill Book Company. 1980.
- *Numerical Methods. 9th Edition.* J. D. Faires y R. L. Burden. Brooks Cole Publishing, 2011.
- *Numerical Computing with Matlab.* C. Moler. 2004.
- *Numerical Analysis. Second* Edition. Walter Gautschi. Birkhauser 2012.

Capítulo 2

Aplicaciones de la interpolación

Una vez que se conoce cómo aproximar una función por un polinomio de interpolación, pues es interesante saber la importancia que tiene contar con una aproximación polinómica para una función $f(x)$, cuando se desea realizar determinadas operaciones sobre dicha función; asunto que será abordado en este capítulo. Específicamente se introducen los procesos de diferenciación e integración aproximada. La diferenciación e integración son conceptos definidos en el análisis mediante procesos de paso al límite y existen reglas para realizar los cálculos prácticos. En los casos en que estas reglas no pueden ser aplicadas hay que recurrir a la resolución aproximada de dichos problemas. Como el proceso del paso al límite no se puede realizar en una computadora, entonces se reemplaza por un proceso finito. Las bases teóricas para realizar los cálculos de derivadas e integrales de forma aproximada descansan sobre la interpolación polinomial. La idea básica es extremadamente simple: en lugar de efectuar la operación sobre la función f, ésta se efectúa sobre un polinomio de interpolación adecuado:

$$f(x) = p(x) \curvearrowright \begin{cases} D(f) = f'(a) \approx D(p) = p'(a) \\ I(f) = \int_a^b f(x)\,dx \approx I(p) = \int_a^b p(x)\,dx \end{cases}$$

Denotemos por L ambos operadores D e I, con lo cual la aproximación de f por p conduce a la aproximación de $L(f)$ por $L(p)$:

$$f(x) \approx p(x) \curvearrowright L(f) \approx L(p)$$

Al estimar el error $L(f) - L(p)$ de dicha aproximación, debido a la linealidad de los operadores diferenciales e integrales, tenemos que si $e(x)$ es el error de la aproximación de f por p , entonces el error de aproximar $L(f)$ por $L(p)$ estará dado por $L(e)$:

$$e(x) = f(x) - p(x) \curvearrowright L(e) = L(f) - L(p)$$

es decir, conocido el error de interpolación $e(x)$, bastará derivarlo o integrarlo para obtener el error de $D(p)$ o $I(p)$ respectivamente.

El empleo del polinomio de interpolación en la diferenciación e integración numéricas, puede considerarse de igual o mayor importancia en la práctica que cuando lo usamos para hallar valores aproximados de una función.

2.1. Derivación numérica

El cálculo exacto de la derivada de una función $f(x)$ en un punto x_0 exige continuidad de la función en dicho punto, ya que se define mediante el proceso de paso al límite

$$f^{'}(x_0) = lim_{h\longrightarrow 0}\frac{f(x_0+h)-f(x_0)}{h} \tag{2.1}$$

Suponiendo que se tenga la expresión analítica de una función continua $f(x)$, para poder realizar el proceso de paso al límite en la computadora éste se debe sustituir por un proceso finito. Si la función viene dada por un conjunto de valores discretos pues tenemos que encontrar primero una expresión analítica para la función que aproxima dichos valores. En ambos casos para calcular la derivada tenemos que acudir a la derivación aproximada.
Para deducir las fórmulas de derivación aproximada, sustituimos f definida sobre el intervalo $[a,b]$ por un polinomio de interpolación $p(x)$ y derivamos el polinomio,

$$f(x) \approx p(x) \curvearrowright f'(x) \approx p'(x),\ x \in [a,b]$$

Análogamente se procede para determinar derivadas de órdenes superiores. Por otra parte si conocemos el error del polinomio de interpolación

$$e(x) = f(x) - p(x)$$

entonces el error de la derivada $p'(x)$ estará dado por

$$e'(x) = f'(x) - p'(x)$$

es decir la derivada del error

$$E(p'(x)) = (e(x))'$$

La derivación numérica es una operación de menor exactitud que la interpolación, pues la coincidencia de las ordenadas $f(x_i)$ y $p(x_i)$ sobre el intervalo $[a,b]$ no garantiza la proximidad de las derivadas $f'(x_i)$ y $p'(x_i)$. Esto se puede representar gráficamente por la diferencia de las pendientes de las tangentes a f y p en cada nodo.

Aproximación para las derivadas. Uso de la fórmula de Newton para nodos equidistantes

Habíamos visto anteriormente que la fórmula de Newton para nodos equidistantes está dada por la expresión

$$P_n(x) = P_n(x_0+sh) = \sum_{i=0}^{n}\Delta^i f_0 \begin{pmatrix} s \\ i \end{pmatrix}$$

donde $s = \frac{x-x_0}{h}$ y los coeficientes $\Delta^i f_0$ son los que encabezan las columnas de la tabla de diferencias finitas construida a partir de los nodos $x_0, x_{1,}...x_n$, siendo $x_i = x_0 + ih$

En forma desarrollada,

$$P_n(x_0+sh) = f_0 + s\Delta f_0 + \frac{s(s-1)}{2!}\Delta^2 f_0 + ... + \frac{s(s-1)...(s-n+1)}{n!}\Delta^n f_0 \tag{2.2}$$

De la relación entre diferencias divididas, derivadas y diferencias finitas,

$$f[x_0, ..., x_k] = \frac{f^{(k)}(\xi)}{k!} = \frac{\Delta^k f_0}{k!h^k} = \frac{\nabla^k f_k}{k!h^k}, \qquad x_0 < \xi < x_k$$

y aproximando la función f por (2.2), se tiene que

$$\frac{df}{dx} \approx \frac{dp}{dx} = \frac{dp}{ds}\frac{ds}{dx} = \frac{1}{h}\frac{dp}{ds}$$

$$\begin{aligned} f'(x) &\approx \frac{1}{h}\left[\Delta f_0 + \frac{2s-1}{2}\Delta^2 f_0 + \frac{3s^2-6s+2}{6}\Delta^3 f_0 + \frac{4s^3-18s^2+22s-6}{24}\Delta^4 f_0 + ...\right] \\ &= p_n'(x_0 + sh). \end{aligned}$$

A partir de la expresión anterior se pueden obtener las derivadas de orden superior.

Para usar estas fórmulas, es necesario fijar n o sea el número $(n+1)$ de nodos y calcular s, que depende del punto x donde se quiera aproximar la derivada y del punto que se escoja como x_0.

El error correspondiente estará dado por la derivada del error de interpolación, es decir por el término $n+1$ de la expresión correspondiente.

Ejemplo 27 *Aproximar la primera derivada con $n = 1$ (o sea con dos nodos de interpolación) en el nodo de la izquierda, $(x = x_0)$.*

$$\begin{aligned} f'(x_0) &\approx \frac{1}{h}\Delta f_0 = \frac{f_1 - f_0}{h} \\ f'(x_0) &\approx \frac{f(x_0 + h) - f(x_0)}{h} \end{aligned}$$

$x = x_0 \curvearrowright s = \frac{x - x_0}{h} = \frac{x_0 - x_0}{h} = 0$

$$\begin{aligned} E(f'(x)) &\curvearrowright E_1(f') \approx \frac{1}{h}\frac{2s-1}{2}\Delta^2 f_0 = \frac{-1}{2h}h^2 f''(\xi) = O(h) \\ &= -\frac{h}{2}f''(\xi), x_0 < \xi < x_1 \end{aligned}$$

Esta aproximación se puede obtener también si se asume que f es una función continua, con derivada continua hasta el orden 2, utilizando una herramienta conocida del Análisis Matemático: la serie de Taylor, veamos:

$$f(x_0 + h) \approx f(x_0) + f'(x_0)h + f''(\xi)\frac{h^2}{2} \tag{2.3}$$

$$\frac{f(x_0 + h) - f(x_0)}{h} - f''(\xi)\frac{h}{2} \approx f'(x_0) \tag{2.4}$$

Ejemplo 28 *Aproximar la segunda derivada con 3 nodos de interpolación ($n = 2$) en el nodo central ($x = x_1$)*

$x = x_1 \curvearrowright s = \frac{x_1 - x_0}{h} = \frac{h}{h} = 1$

$$\begin{aligned} f''(x_1) &\approx \frac{1}{h^2}\Delta^2 f_0 = \frac{1}{h^2}\Delta(\Delta f_0) \\ &\approx \frac{1}{h^2}\Delta(f_1 - f_0) = \frac{1}{h^2}[\Delta f_1 - \Delta f_0] \\ &= \frac{f_0 - 2f_1 + f_2}{h^2} \end{aligned}$$

diferencia finita central para la segunda derivada

$E(f''(x)) \curvearrowright E_2(f'') \approx \frac{1}{h^2}\left[(s-1)\Delta^3 f_0 + \frac{6s^2-18s+11}{12} + \Delta^4 f_0\right] = -\frac{1}{12}h^2 f^{iv}(\xi) = O(h^2) \quad x_0 < \xi < x_2$

Esta aproximación para la segunda derivada también se puede obtener sumando los desarrollos de Taylor de $f(x+h)$ y $f(x-h)$, truncando después del tercer término y dividiendo por h^2.

Ejemplo 29 *Aproximar la primera derivada en $x = x_0$ con $n = 2$, con nodos de interpolación x_0, x_1, x_2*

$x = x_0 \curvearrowright s = 0$

$$\begin{aligned} f'(x_0) &\approx \frac{1}{h}\left[\Delta f_0 + \frac{2s-1}{2}\Delta^2 f_0\right] = \frac{1}{h}\left[(f_1 - f_0) - \frac{1}{2}(f_0 - 2f_1 + f_2)\right] \\ &= \frac{1}{2h}[-3f_0 + 4f_1 - f_2] \end{aligned}$$

$E(f'(x)) \curvearrowright E_2(f') \approx \frac{1}{h}\left[\frac{3s^2-6s+2}{6}\Delta^3 f_0\right] = \frac{h^2}{3}f'''(\xi) = O(h^2)$

Note que en este caso la combinación adecuada para usar los desarrollos en serie de Taylor no es fácil de deducir, sin embargo la función de Newton nos da un método general para aproximar derivadas.

Una de las aplicaciones más importantes de las aproximaciones para las derivadas está en su uso en la discretización de ecuaciones diferenciales; específicamente en el tan popular Método de Diferencias Finitas (MDF). Por ejemplo si consideramos un caso muy simple del problema de fronteras para la llamada ecuación de la conducción del calor estacionaria en una dimensión,

$$\begin{aligned} -u''(x) + u(x) = f(x),\ \forall x \in (0,1) \\ u(0) = \alpha,\ u(1) = \beta. \end{aligned} \tag{2.5}$$

Considerando una discretización con nodos equidistantes $0 = x_0 < x_1 < ... < x_n = 1$ del intervalo $[0,1]$, sustituyendo la segunda derivada por su aproximación mediante diferencia finita central en los nodos interiores y a las funciones $u(x)$ y $f(x)$ por sus valores en los nodos x_i, $i = 1, .., n-1$, se obtiene

$$\begin{aligned} -\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} + u_i = f_i,\ i = 1, ..., n-1 \\ u(x_0) = u_0 = \alpha,\ u(1) = u_n = \beta. \end{aligned} \tag{2.6}$$

Ahora, en lugar de buscar una expresión analítica para $u(x)$ como solución del modelo (2.5), se buscan valores discretos $u(x_i)$ como solución del sistema de ecuaciones lineales

$$-u_{i-1} + (2 + h^2)u_i - u_{i+1} = h^2 f_i,\ i = 1, ..n-1, \tag{2.7}$$

para más detalles sobre el Método de Diferencias Finitas ver [J. W. Thomas. Numerical Partial Differencial Equations: Finite Difference Methods. Springer Verlag New York, second edition 1998].

2.2. Integración aproximada

En esta sección nos ocupa el problema de tener que calcular la integral de una función $f(x)$, definida en el intervalo $[a, b]$

$$I(f) = \int_a^b f(x)\, dx$$

En el Análisis Matemático, el concepto de integral es también definido mediante un proceso de paso al límite: el valor $I(f)$ existe, si

$$lim_{n\longrightarrow\infty} \sum_{i=1}^{n} f(\xi_i)\Delta x_i \tag{2.8}$$

existe para algún ξ_i y para algún Δx_i cuando $\Delta x_i \longrightarrow 0$. Para el cálculo en la computadora, este proceso debe ser sustituido por un proceso finito. Se puede pensar intuitivamente en ir calculando las sumas parciales $S_n = \sum_{i=1}^{n} f(\xi_i)\Delta x_i$ cuando $n \to \infty$. Sin embargo la convergencia de S_n a $I(f)$ es muy lenta, por lo que esta vía no se usa en la práctica.

Además, se pueden presentar otras situaciones que impiden el cálculo de una integral de manera exacta, como puede ser que:

1. aunque se conozca la primitiva de f, ésta no sea expresable en términos de un número finito de funciones elementales, por ejemplo, $\int_a^b e^{-x^2} dx$,

2. no se tiene una expresión analítica de f, sino que sólo se conocen sus valores en un número finito de puntos.

Ante estas situaciones nos vemos obligados a emplear métodos numéricos para calcular aproximadamente el valor de la integral $I(f)$. El problema de la integración numérica o cuadratura numérica formulado de manera general consiste en estimar el valor de

$$I(f) = \int_a^b f(x)\, dx$$

como

$$\int_a^b f(x)w(x)dx = \sum_{i=1}^{n} f(x_i)w_i + E_n(f) \tag{2.9}$$

donde w es una función llamada de peso, no negativa e integrable sobre (a, b). Si f es un polinomio de grado menor o igual que d, se dice que la fórmula de cuadratura es exacta de orden d cuando $E_n(f) = 0, \forall f \in \mathbb{P}_d$. La expresión (2.9) se llama interpolatoria si es exacta para polinomios de orden $n-1$. Las fórmulas interpolatorias son precisamente las que se obtienen al sustituir la función en el integrando por un polinomio de interpolación, esto es

$$\sum_{i=1}^{n} w_i f(x_i) = \int_a^b p_{n-1}(f, x_1, ..., x_n; x)w(x)dx$$

donde $p_{n-1}(x)$ dado en la forma de Lagrange,

$$p_{n-1}(x) = \sum_{i=1}^{n} f(x_i) l_i(x)$$

$l_i(x)$ son las funciones base de Lagrange y $w_i = \int_a^b l_i(x) w(x) dx$.

2.2.1. Fórmulas de Newton Cotes o fórmulas de tipo interpolatorio

Sea f una función suave en el intervalo $[a, b]$ y $p_n(x)$ el polinomio de interpolación de grado menor o igual que n que aproxima a f en los nodos $x_0, x_1, ..., x_n$ que pertenecen a $[a, b]$. Lo más sencillo es tomar nodos equidistantes y considerar la expresión dada por la fórmula de Newton en diferencias finitas hacia adelante :

$$p_n(x_0 + sh) = f_0 \begin{pmatrix} s \\ 0 \end{pmatrix} + \Delta f_0 \begin{pmatrix} s \\ 1 \end{pmatrix} + \Delta^2 f_0 \begin{pmatrix} s \\ 2 \end{pmatrix} + ... + \Delta^n f_0 \begin{pmatrix} s \\ n \end{pmatrix}$$

Al aproximar la integral de $f(x)$ por la integral de $p_n(x)$, considerando diferentes órdenes n para el polinomio de interpolación, se obtienen diferentes fórmulas para aproximar el valor de la integral, conocidas como las fórmulas de cuadratura de Newton-Cotes o fórmulas de tipo interpolatorio.

Se considera una partición del intervalo $[a, b]$; $a = x_0 < x_1 < ... < x_n = b$ y se plantea la integral entre $a = x_0$ y $b = x_n$, sustituyendo a la función $f(x)$ por el polinomio de interpolación de grado n:

$$\begin{aligned} I(f) &= \int_a^b f(x)\, dx \approx I(p) = \int_{x_0=a}^{x_n=b} p_n(x)\, dx \\ &= \int_0^n \left\{ f_0 + \Delta f_0 \begin{pmatrix} s \\ 1 \end{pmatrix} + \Delta^2 f_0 \begin{pmatrix} s \\ 2 \end{pmatrix} + ... + \Delta^n f_0 \begin{pmatrix} s \\ n \end{pmatrix} \right\} h ds \end{aligned} \tag{2.10}$$

donde $x = x_0 + sh$, $dx = hds$, $x = x_0 \curvearrowright s = 0$, $x = x_n \curvearrowright s = n$. El error de esta aproximación está dado por la integral del error de interpolación:

$$E\left(\int p_n(x)\right) = I(f(x)) - I(p_n(x)) = I(f(x) - p_n(x)) = I(e_n(x)) \approx \int_0^n \Delta^{n+1} f_0 \begin{pmatrix} s \\ n+1 \end{pmatrix} h ds$$

Las fórmulas de integración numérica que se obtienen a partir de la expresión (2.10) supuesto que los nodos son equidistantes:

$$x_{i+1} - x_i = h = cte, \quad 0 \leq i \leq n-1$$

se denominan fórmulas de Newton-Cotes de tipo cerrado, ya que se usan los valores de la función en los extremos del intervalo de integración. Los casos particulares más conocidos son los que se obtienen para $n = 1$ (regla de los trapecios) y para $n = 2$ (fórmula de Simpson), es decir, cuando se aproxima a f por una recta y por una parábola de interpolación en los subintervalos $[x_{i-1}, x_i]$ y $[x_{2i-2}, x_{2i}]$, respectivamente.

2.2.2. Reglas básicas y compuestas de los trapecios y de Simpson

Fórmula de los trapecios

Para la regla básica de los trapecios, se considera $n = 1$ en $[x_{i-1}, x_i]$, para nodos equidistantes $h = x_i - x_{i-1} \ \ \forall i$ y $x = x_{i-1} + sh$; se obtiene:

$$\begin{aligned} I(p_1) &= \int_0^1 \left[f_{i-1} + \Delta f_{i-1} \begin{pmatrix} s \\ 1 \end{pmatrix} \right] h ds \\ &= h f_{i-1} \int_0^1 ds + h(f_i - f_{i-1}) \int_0^1 s ds \\ &= h f_{i-1} s \Big|_0^1 + h(f_i - f_{i-1}) \frac{s^2}{2} \Big|_0^1 \\ &= h f_{i-1} + h(f_i - f_{i-1}) \frac{1}{2} \\ I(p_1) &= \frac{h}{2}(f_{i-1} + f_i) \end{aligned} \tag{2.11}$$

El error de método en cada subintervalo $[x_{i-1}, x_i]$ estará dado por:

$$\begin{aligned} E(p_1) &= \int e(p_1)\, dx \approx \int_0^1 \Delta^2 f_{i-1} \begin{pmatrix} s \\ 2 \end{pmatrix} h ds \\ &= h \Delta^2 f_{i-1} \int_0^1 \frac{s(s-1)}{2} ds \\ &= \frac{-h}{12} \Delta^2 f_{i-1} \end{aligned}$$

y teniendo en cuenta la relación entre diferencias finitas y derivadas $\Delta^k f_0 = h^k f^{(k)}(\xi)\,,\ \xi \in (x_0, x_k)$ se obtiene,

$$E(p_1) \approx f''(\xi) \frac{-h^3}{12} = O\left(h^3\right), \quad \xi \in (x_{i-1}, x_{i+1})\,.$$

La expresión anterior significa que la fórmula de los trapecios es exacta para polinomios de primer orden.

Para obtener la fórmula compuesta se divide el intevalo de integración $[a, b]$ en m partes iguales y aproximando f por una recta de interpolación en cada subintervalo $[x_{i-1}, x_i]$ se obtiene:

$$\int_{x_0=a}^{x_m=b} f(x)\, dx \sim I(p_1) = \sum_{i=1}^{m} \int_{x_{i-1}}^{x_i} p_1(x)\, dx = \sum_{i=1}^{m} \frac{h}{2}(f_{i-1} + f_i)$$

que es la regla compuesta de los trapecios:

$$I(p_1) = \sum_{i=1}^{m} \frac{h}{2}(f_{i-1} + f_i) = \frac{h}{2}\left[f_0 + 2f_1 + 2f_2 + ... + 2f_{m-1} + f_m\right]. \tag{2.12}$$

Para calcular el error de la regla compuesta de los trapecios pues

$$E_{TC} = -\frac{1}{12}h^3 f''(\xi_0) - \frac{1}{12}h^3 f''(\xi_1) - ... - \frac{1}{12}h^3 f''(\xi_{m-1}) \tag{2.13}$$

como $h = \frac{b-a}{m}$

$$\begin{aligned} E_{TC} &= -\frac{h^2}{12}(b-a)\sum_{i=0}^{m-1} f''(\xi_i) = -\frac{h^2}{12}(b-a)f''(\xi) \\ &= O\left(h^2\right) \end{aligned} \tag{2.14}$$

con $\xi \in [a,b]$. Para obtener la expresión en (2.14) se aplica la siguiente generalización del teorema del valor medio.

Teorema 30 *Sea $f(x)$ una función continua en $[a,b]$ y sean $x_1, ..., x_n$ puntos en $[a,b]$ y sean $g_1, ..., g_n$ números reales todos del mismo signo. Entonces*

$$\sum_{i=1}^{n} f(x_i)g_i = f(\xi)\sum_{i=1}^{n} g_i \ , \ \xi \in [a,b] \tag{2.15}$$

Fórmula de Simpson

Para la regla básica de Simpson se considera $n = 2$ en $[x_{2i-2}, x_{2i}]$, es decir, f se aproxima por una parábola de interpolación con nodos equidistantes $x_{2i-2}, x_{2i-1}, x_{2i}$ y $x = x_{2i-2} + sh$ se obtiene:

$$\begin{aligned} \int_{x_{2i-2}}^{x_{2i}} p_2(x)dx &= \int_0^2 \left[f_{2i-2} + \Delta f_{2i-2}s + \frac{\Delta^2 f_{2i-2}s(s-1)}{2!} \right] hds \\ &= \left[f_{2i-2}s + \Delta f_{2i-2}\frac{s^2}{2} + \Delta^2 f_{2i-2}\frac{s^3}{6} - \Delta^2 f_{2i-2}\frac{s^2}{4} \right] h|_0^2 \\ &= \frac{h}{3}\left[f_{2i-2} + 4f_{2i-1} + f_{2i} \right] \end{aligned}$$

En este caso para calcular el orden de método hay que utilizar el término que contiene $\Delta^4 f_{2i-2}$ ya que

$$\int_0^2 \left[\Delta^3 f_{2i-2}\frac{s(s-1)(s-2)}{3!} \right] hds = 0 \tag{2.16}$$

Por tanto

$$\begin{aligned} E(p_2) &= \int_0^2 \left[\Delta^4 f_{2i-2}\frac{s(s-1)(s-2)(s-3)}{4!} \right] hds \\ &= \frac{-h\Delta^4 f_{2i-2}}{90} \\ &= \frac{-h^5 f^{(iv)}(\xi)}{90}, \ \xi \in (x_{2i-2}, x_{2i+1}) \end{aligned}$$

Análogamente, dividiendo el intervalo $[a,b]$ en un número par $2m$ de partes iguales, y aproximando f por una parábola de interpolación en cada subintervalo $[x_{2i-2}, x_{2i}]$, se obtiene la regla compuesta de Simpson,

$$\int_{x_0=a}^{x_{2m}=b} f(x)\,dx \sim I(p_2) = \sum_{i=1}^{m} \int_{x_{2i-2}}^{x_{2i}} p_2(x)\,dx = \sum_{i=1}^{m} \frac{h}{3}\left[f_{2i-2} + 4f_{2i-1} + f_{2i}\right]$$

$$\begin{aligned} I(p_2) &= \frac{h}{3}\left[f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + ..,4f_{2m-1} + f_{2m}\right] \\ &= \frac{h}{3}\left[E + 4I + 2P\right] \end{aligned}$$

donde $E = f_0 + f_{2m}$, $I = \sum_{i=1}^{m} f_{2i-1}$, $P = \sum_{i=1}^{m-1} f_{2i}$

El error de la regla compuesta es

$$E_{SC} = \frac{(b-a)h^5}{(2h)90} f^{(iv)}(\xi) = \frac{h^4}{180}(b-a) f^{(iv)}(\xi). \tag{2.17}$$

Para las fórmulas de trapecios y Simpson se han obtenido expresiones del error global en todo el intervalo de integración, para los trapecios

$$I(f) - I(p_1) = -\frac{b-a}{12} h^2 f''(\xi) = O\left(h^2\right) \tag{2.18}$$
$$x_{i-1} < \xi < x_i$$

y para la fórmula de Simpson

$$I(f) - I(p_2) = -\frac{b-a}{180} h^4 f^{(iv)}(\xi) = O\left(h^4\right) \tag{2.19}$$
$$x_{2i-2} < \xi < x_{2i}.$$

Estas expresiones requieren para su cálculo la evaluación de $f^{(s)}(\xi)$ que generalmente es difícil, si no imposible. No obstante, el conocimiento de tal expresión para el error de método nos indica la velocidad con que el valor aproximado de la integral converge al valor exacto de la integral, $I(p) \to I(f), h \to 0$, lo cual expresamos como $O(h^r)$. Sin embargo en la práctica se necesita poder estimar el error con que se aproxima el valor de la integral, lo que se verá más adelante.

2.2.3. Fórmulas de cuadratura gaussiana

La teoría de las fórmulas de cuadratura gaussianas se basa en la teoría de polinomios ortogonales.

Definición 31 *Una sucesión de polinomios $\{p_j\}_{j=0}^{\infty}$ con $p_n(x)$ de grado n, se llamará sucesión de polinomios ortogonales respecto al producto escalar*

$$(f,g) := \int_a^b f(x)g(x)dx \tag{2.20}$$

si se verifica $(p_j, p_k) = 0$, $j \neq k$.

Las fórmulas de integración numérica se expresan como una combinación lineal, con pesos ω_i, de valores de la función en nodos del intervalo de integración seleccionados adecuadamente,

$$\int_a^b f(x)dx \approx \omega_1 f(x_1) + ... + \omega_n f(x_n). \tag{2.21}$$

El siguiente teorema relaciona las fórmulas de Newton-Cotes con esta forma de expresar la integración numérica.

Teorema 32 *Dados los nodos $x_1, ..., x_n$ la fórmula de cuadratura*

$$\int_a^b f(x)dx \approx \omega_1 f(x_1) + ... + \omega_n f(x_n). \tag{2.22}$$

es de tipo interpolatorio si y sólo si es exacta para los polinomios $1, x, ..., x^{n-1}$.

Es decir una fórmula de tipo interpolatorio con n nodos tiene al menos orden de precisión $n-1$. El orden de precisión óptimo usando n nodos es $2n-1$ y se obtiene para las fórmulas de cuadratura gaussianas.

Teorema 33 *Sean $x_1, x_2, ..., x_n$ las raíces del polinomio ortogonal $p_n(x)$ para la medida $d\omega(x)$ en (a, b). Supongamos que se hallan los pesos $\omega_1, \omega_2, ...\omega_n$ imponiendo la exactitud para los polinomios de grado menor o igual que $n-1$, es decir que se construye la fórmula de tipo interpolatorio*

$$\int_a^b f(x)dx \approx \omega_1 f(x_1) + ... + \omega_n f(x_n). \tag{2.23}$$

Entonces dicha fórmula tiene orden de precisión $2n-1$.

2.2.4. Estimación del error de método por doble cómputo

Las expresiones obtenidas para el error global en las fórmulas de Newton-Cotes tienen la forma general:

$$E_h = I(f) - I_h(p) = ch^r f^{(s)}(\xi) = O(h^r) \tag{2.24}$$

donde c: constante, r, s :números naturales, h : paso de integración, $\xi = \xi(h)$ es un punto desconocido del intervalo de integración. La información que nos brindan estas expresiones puede usarse para:

1. estimar el error del valor aproximado $I_h(p)$ obtenido con paso h, sin necesidad de evaluar $f^{(s)}(\xi)$, lo cual mostraremos aplicando el llamado método de doble cómputo, que como su nombre indica se trata de usar dos cálculos. En específico se calculan dos valores aproximados de la integral $I(f)$, uno con paso de integración $2h$ que denotaremos por $I_{2h}(p)$, y otro con paso de integración h, $I_h(p)$. Para comparar ambas aproximaciones en el intervalo $[x_{i-2}, x_i]$ aplicamos la regla de los trapecios con paso $2h$, una sola vez y con paso h, la aplicamos dos veces.

2. calcular aproximaciones más precisas de $I(f)$ mediante extrapolación, lo cual utilizaremos para deducir el algoritmo de integración numérica de Romberg.

Veamos como proceder con el método de doble cómputo usando como regla de integración la fórmula de los trapecios. Supongamos que la función segunda derivada f'' es suficientemente suave en $[x_{i-2}, x_i]$. Podemos admitir entonces que en la expresión del error para el método de los trapecios, $-\frac{b-a}{12}h^2 f''(\xi)$,

$$-\frac{1}{12}f''(\xi_1) \approx -\frac{1}{12}f''(\xi_2) = k$$

luego para los errores de método tendremos las expresiones:

$$I(f) - I_{2h}(p_1) \approx (2h)^2 k + O\left(h^2\right) \tag{2.25}$$

$$I(f) - I_h(p_1) \approx (h)^2 k + O\left(h^2\right) \tag{2.26}$$

Restando (2.26) de (2.25) y despejando h^2k

$$h^2 k \approx \frac{I_h(p_1) - I_{2h}(p_1)}{3}. \tag{2.27}$$

Sustituyendo (2.27) en (2.26), obtenemos una estimación para el error que se comete al calcular el valor aproximado de la integral por la fórmula compuesta de los trapecios con paso más pequeño $I_h(p_1)$, en $[x_{i-2}, x_i]$:

$$E_h(p_1) = I(f) - I_h(p_1) \approx \frac{1}{3}\left[I_h(p_1) - I_{2h}(p_1)\right] \tag{2.28}$$

Observe que (2.28) nos da una estimación de E_h en términos de I_he I_{2h}, en la cual no aparece $f''(\xi)$. En general, si el error de método de la aproximación $I_h(p)$ es proporcional a h^r, entonces

$$E_h(p) = I(f) - I_h(p) \approx \frac{1}{2^r - 1}\left[I_h(p) - I_{2h}(p)\right] \tag{2.29}$$

Las expresiones (2.28) y (2.29), además de proporcionarnos una estimación del error E_h, nos permiten obtener una aproximación de $I(f)$ más precisa, que llamaremos valor extrapolado y denotaremos $I_0(p)$. La obtención de este valor es posible gracias al llamado método de extrapolación de Richardson que explicitamos a continuación.

2.2.5. Extrapolación de Richardson

El método de extrapolación de Richardson, desarrollado por Lewis Fry Richardson (1881-1953), permite construir a partir de una sucesión convergente otra sucesión que converge más rápidamente. Esta técnica se usa frecuentemente para mejorar los resultados de métodos numéricos a partir de una estimación previa.

Despejando $I(f)$ en (2.29) obtenemos

$$I(f) \approx I_h(p) + \frac{1}{2^r - 1}\left[I_h(p) - I_{2h}(p)\right] = I_0(p)$$

Los valores extrapolados obtenidos a partir de los trapecios:

$$I_0(p_1) = \frac{1}{3}\left[4I_h(p_1) - I_{2h}(p_1)\right] \tag{2.30}$$

y a partir de una fórmula general de integración numérica de orden r :

$$I_0(p) = \frac{1}{2^r - 1}\left[2^r I_h(p) - I_{2h}(p)\right] \tag{2.31}$$

son más precisos que los correspondientes $I_h(p_1)$ e $I_h(p)$, porque se obtienen sumando la estimación de error al valor aproximado calculado:

$$I_0(p) = I_h(p) + E_h(p)$$

La expresión ¨valor extrapolado ¨ se justifica porque a partir de valores aproximados I_{2h} e I_h estamos estimando el valor $I_{0,}$ que es una aproximación de $I(f)$ para $h = 0$, fuera del intervalo $[h, 2h]$ Este algoritmo de extrapolación se usa para construir el algoritmo de integración numérica de Romberg.

2.2.6. Algoritmo de Romberg

Combinando la fórmula de integración numérica de los trapecios con extrapolación de Richardson se obtiene lo que se conoce como el algoritmo de Romberg. Para ello es necesario disponer de valores aproximados de la integral $I_{2h}(p_1), I_h(p_1), I_{\frac{h}{2}}(p_1), ...$ obtenidos por subdivisión sucesiva del paso de integración a la mitad, o lo que es igual, por duplicación sucesiva del número de subintervalos del intervalo de integración $[a, b]$. Nos interesa entonces obtener el valor $I_h(p_1)$ a partir de $I_{2h}(p_1)$, sin repetir la evaluación de f en los puntos ya considerados para evaluar $I_{2h}(p_1)$; para lo cual se suma a I_{2h} el incremento que se obtiene evaluando f en los puntos intermedios que aparecen al dividir el paso a la mitad: $I_h = g(I_{2h}) + \Delta$. El objetivo inmediato es obtener una expresión de la regla de los trapecios en forma recurrente, que sea válida también cuando a partir de I_h se quiera calcular $I_{\frac{h}{2}}$, y así sucesivamente reduciendo el paso a la mitad.

Regla recurrente de los trapecios

Sea $[a, b]$ el intervalo de integración. Al aplicar la regla de los trapecios para calcular el valor aproximado de la integral se obtiene

$$I_h(p_1) = (b - a)\left[\frac{1}{2}f(a) + \frac{1}{2}f(b)\right] \tag{2.32}$$

Entonces para hacer una estimación del error dividimos el paso a la mitad y aplicamos de nuevo trapecios, ahora con dos intervalos y se obtiene

$$I_{\frac{h}{2}}(p_1) = \frac{b - a}{2}\left[\frac{1}{2}f(a) + \frac{1}{2}f(b) + f\left(a + \frac{b - a}{2}\right)\right] \tag{2.33}$$

Como se observa los dos primeros sumandos fueron calculados al realizar la aproximación con la regla básica. Para actualizarlo, solo tendríamos que multiplicar por $\frac{1}{2}$. De ahí que si denotamos $I_h(p_1)$ como T_0^0 y a $I_{\frac{h}{2}}(p_1)$ como T_1^0, donde $h = b - a$, entonces

$$T_1^0 = \frac{1}{2}T_0^0 + \frac{b - a}{2}f\left(a + \frac{b - a}{2}\right) \tag{2.34}$$

De forma análoga si dividimos de nuevo el paso a la mitad $(\frac{h}{4})$ y aplicamos trapecios tendremos $I_{\frac{h}{4}}(p_1) = T_2^0$, con

$$T_2^0 = \frac{b-a}{4}\left[\frac{1}{2}f(a) + \frac{1}{2}f(b) + f\left(a + \frac{b-a}{4}\right) + f\left(a + 2\frac{b-a}{4}\right) + f\left(a + 3\frac{b-a}{4}\right)\right]. \quad (2.35)$$

Como se observa en T_2^0 hay sumandos que ya fueron calculados en T_1^0: el valor de f en los extremos del intervalo y en los nodos de índice par si comenzamos a numerar con cero, por lo que podemos simplicar la expresión ,

$$T_2^0 = \frac{1}{2}T_1^0 + \frac{b-a}{4}\left[f\left(a + \frac{b-a}{4}\right) + f\left(a + 3\frac{b-a}{4}\right)\right]. \quad (2.36)$$

Formalizando la notación. Sea $n = 2^N$ $(N = 0, 1, 2, ..)$ el número de divisiones sucesivas del intervalo de integración $[a, b]$. Denotemos por $I_{2h}(p_1) = T_{N-1}^0$: valor aproximado de $I(f)$ calculado por la regla de los trapecios con paso $2h = \frac{b-a}{2^{(N-1)}}$, $I_h(p_1) = T_N^0$: valor aproximado de $I(f)$ calculado por la regla de los trapecios con paso $h = \frac{b-a}{2^N}$, entonces se obtiene la llamada regla recurrente de los trapecios:

$$T_N^0 = \frac{1}{2}T_{N-1}^0 + h\sum_{i=1}^{2^{(N-1)}} f(a + (2i-1)h), N \geq 1 \quad (2.37)$$

donde se ve que la función f se evalúa $n + 1 = 2^N + 1$ veces para hallar T_N^0, independientemente de que antes, se hayan obtenido o no, los valores $T_0^0, T_1^0, ..., T_{N-1}^0$. Veamos qué ventajas nos reporta esta expresión obtenida para la regla de los trapecios.

Aplicación de la extrapolación de Richardson

Anteriormente vimos que usando la regla de los trapecios a partir de dos aproximaciones $I_h(p_1)$ e $I_{2h}(p_1)$ de $I(f)$ con pasos h y $2h$ respectivamente es posible obtener un valor extrapolado más preciso,

$$I_0(p_1) = \frac{4I_h(p_1) - I_{2h}(p_1).}{3}$$

Utilizando la notación introducida:

$$I_h(p_1) = T_N^0 \quad \text{y} \quad I_{2h}(p_1) = T_{N-1}^0$$

y denotando por T_N^1 el valor extrapolado calculado a partir de T_N^0 y T_{N-1}^0, obtenemos

$$T_N^1 = \frac{4T_N^0 - T_{N-1}^0}{3}, \quad N \geq 1. \quad (2.38)$$

Por ejemplo, para $N = 1$ obtenemos

$$\begin{aligned} T_1^1 &= \frac{4T_1^0 - T_0^0}{3} \\ &= \frac{h}{3}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] \end{aligned} \quad (2.39)$$

Como se observa (2.39) es la regla básica de Simpson, con $\frac{b-a}{2} = h$ en el intervalo $[a, b]$. Se puede demostrar que la sucesión $\{T_N^1\}$, calculada a partir de T_{N-1}^0 y T_N^0 coincide con la aproximación de

$I(f)$ que podría ser calculada por la fórmula de Simpson con $N = 2^n$ subintervalos. Entonces el error global de T_N^1 coincide con el de Simpson, es decir, $O(h^4)$.

Si $f^{(IV)}$ es continua y acotada en $[a, b]$

$$I(f) - T_N^1 \underset{N\to\infty}{\to} 0$$

luego la sucesión $\{T_N^1\}$ converge al verdadero valor de la integral cuando $N \to \infty$.

Utilizando la extrapolación de Richardson nuevamente con dos elementos consecutivos de la sucesión $\{T_N^1\}$ se obtiene análogamente

$$T_N^2 = \frac{16T_N^1 - T_{N-1}^1}{15}, \text{ para } N \geq 2 \tag{2.40}$$

pues en este caso $r = 4$ (Potencia de h en la expresión del error global de la fórmula de Simpson).

Investigando la sucesión $\{T_N^2\}$ se ve que coincide con la aproximación de $I(f)$ que podría haberse calculado mediante la fórmula de integración de Newton-Cotes para 4 puntos (polinomio de grado 3), que tiene error global $O(h^6)$.

Si reiteramos las extrapolaciones, ahora con $r = 6$, obtenemos

$$T_N^3 = \frac{64T_N^2 - T_{N-1}^2}{63}, \text{ para } N \geq 3 \tag{2.41}$$

La fórmula que sirve de base para la extrapolación en el método de Romberg puede escribirse de manera general como:

$$T_N^j = \frac{4^j T_N^{j-1} - T_{N-1}^{j-1}}{2^{2j} - 1}, \text{ para } N \geq j \tag{2.42}$$

donde $j = \frac{r}{2}$, ya que r toma valores pares, $2, 4, ..,$ y entonces $2^r = 2^{2j} = 4^j$.

Tabla de Romberg

Las sucesiones $\left\{T_N^j\right\}$ pueden disponerse en una tabla en la siguiente forma:

j=0	j=1	j=2	$\cdots$
T_0^0			
T_1^0	T_1^1		
T_2^0	T_2^1	T_2^2	
$\vdots$	$\vdots$	$\vdots$	$\ddots$
T_{N-2}^0			
T_{N-1}^0	T_{N-1}^1	T_{N-1}^2	
T_N^0	T_N^1	T_N^2	$\cdots$

Observe que la tabla de Romberg se calcula en forma análoga a una tabla de diferencias divididas. Es importante señalar que para un eficiente desempeño del algoritmo lo ideal sería saber cuántas filas hay que calcular para obtener la mejor aproximación de la integral. Sin embargo este concepto de mejor aproximación es ambiguo, ya que depende del usuario y de la aplicación, por tanto lo que se hace en la práctica es fijar un umbral ε para el error con que se quiere obtener la aproximación de la integral. Se calculan nuevas filas (es decir se realizan nuevas subdivisiones del intervalo) hasta que la diferencia entre dos elementos consecutivos de la diagonal sea menor ó igual que dicho umbral, es decir $|T_N^N - T_{N-1}^{N-1}| \leq \varepsilon$. Muchas veces se calcula una fila más, para tener además $|T_{N-1}^{N-1} - T_{N-2}^{N-2}| \leq \varepsilon$, ver (Burden and Faires).

Cómo se usa entonces la extrapolación de Richardson para obtener las sucesiones de Romberg?

Algoritmo de Romberg

Dada la función $f(x)$, definida en $[a,b]$ por su expresión analítica:

Algoritmo 34

Poner $b-a \to h$
Calcular $\frac{h}{2}\left[f(a)+f(b)\right] \to T_0$
Para $n = 1, 2, ..$
$\frac{h}{2} \to n$
calcular $\frac{1}{2}T_{n-1} + h\sum_{i=1}^{2^{n-1}} f\left(a+(2i-1)h\right) \to T_n$
$n \to k$
Para $j = 1, 2, .., n$*:*
calcular $\left(4^j T_k - T_{k-1}\right)/\left(4^j-1\right) \to T_{k-1}$
si $\left|\frac{T_{k-1}-T_k}{T_K}\right| < \varepsilon$,
parar e imprimir
$k-1 \to k$

La simplicidad y rápida convergencia del algoritmo de Romberg hacen que aventaje a otros métodos, ya que para una precisión determinada, el número n de subdivisiones del intervalo $[a,b]$ que se requiere es mucho menor, lo cual tiene como consecuencia la minimización de la acumulación de errores de redondeo. Debido a estas ventajas del método de Romberg, la regla de los trapecios es generalmente más útil que todas las fórmulas de integración numérica de orden superior.

Ejemplo 35 *Calcular* $\int_1^5 \frac{1}{x}dx$ *por el algoritmo de Romberg.*

$b-a=4$, $f(x)=\frac{1}{x}$

h	N	$n=2^n$	T_N^0	T_N^1	T_N^2	T_N^3
4	0	1	2,4			
2	1	2	1,86667	1,68889		
1	2	4	1,68334	1,62963	1,62567	
0,5	3	8	1,62897	1,61085	1,60960	1,60934

$I(f) \approx T_3^3 = 1{,}60934$
(4 cifras significativas correctas, pues $\delta \approx 0{,}0001 < 5\times 10^{-4}$)

Valor exacto:

$$I(f) = \int_1^5 \frac{1}{x}dx = \ln x\big|_1^5 = \ln 5 - \ln 1$$
$$= 1{,}6094379$$

luego el resultado de T_3^3 coincide con $I(f)$ en 4 cifras significativas. Si se usara la regla de los trapecios sin hacer extrapolación, el valor más preciso que se obtiene es

$$T_3^{0,} = 1{,}62897, \text{ para } n = 8 \ \ ó \ \ h = 0{,}5,$$

que sólo tiene 2 cifras coincidentes.

Si se usara la fórmula de Simpson, el valor más preciso que se obtiene es:

$$T_3^{1,} = 1{,}61085, \text{para } N = 8 \ \ ó \ \ h = 0{,}5,$$

que también tiene dos cifras coincidentes. Luego para obtener la precisión alcanzada en el método de Romberg con $N = 8$, habría que tomar N mucho mayor si se usaran dichos métodos.

2.3. Ejercicios para el estudio independiente

1. Una de las expresiones más usadas para aproximar la derivada $f'(x_0)$ de una función $f \in C^2[a,b]$ es la conocida como diferencias centradas:

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

   a) ¿Por qué cree que se conoce con ese nombre?

   b) ¿Qué error se comete al aproximar la derivada de una función utilizando las diferencias centradas?

   c) ¿Por qué no se puede aproximar la derivada de una función mediante diferencias centradas con un valor de h *"tan pequeño como se quiera"*?

   d) Tomando en cuenta que el error que se comete depende del parámetro h, y que no puede ser *"tan pequeño como se quiera"*, ¿cuál es el valor de h para el cual se comete el menor error posible?

2. Ahora que ya sabe aproximar la derivada de funciones de una variable real:

   a) Diga cómo se puede implementar, en un lenguaje de programación cualquiera, el cálculo aproximado del gradiente de una función $f : \mathbb{R}^n \to \mathbb{R}$.

   b) Implemente su propuesta en el lenguaje de su preferencia.

   c) Dado un conjunto de vectores de $\mathbb{R}^n$ $\{d_1, d_2, \ldots, d_n\}$, una función $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, y un punto $x_0 \in \mathbb{R}^n$, use su implementación para verificar cuáles de los vectores d_i son direcciones de descenso para $f(x)$ en el punto x_0.

3. Una forma de aproximar la derivada de orden k de una función cualquiera $f \in C^\infty[a,b]$, es construir un polinomio de interpolación de grado n que interpole a $f(x)$ en $n-1$ puntos y derivar el polinomio k veces.

   a) Si se desea obtener una aproximación de la derivada k-ésima con un error $O(h^m)$, ¿en cuántos puntos, cómo máximo, se debe interpolar la función f?

4. Dados $k, q, L \in \mathbb{R}$, se desea hallar una función $y \in C^5[0, L]$ tal que:

$$y^{(IV)} + ky = q, \quad x \in [0, L]$$
$$y(0) = y'(0) = 0$$
$$y(L) = y''(L) = 0,$$

donde $y^{(VI)}$ es la derivada de orden 4 de la función y.

   a) Implemente una función que reciba como argumentos los valores de k, q, L, y un número N que indique en cuántos puntos particionar el intervalo $[0, L]$; y devuelva los valores de la función y en los puntos $t_i = i\frac{L}{N}$, utilizando el método de diferencias finitas.

   b) Experimente con diferentes valores de k, q y N para explorar su influencia en el gráfico de la función y.

5. Utilizando el desarollo en serie de Taylor de $f(x+h)$, y $f(x-h)$, y sumándolos y restándolos ingeniosamente para despejar $f'(x)$:

   a) Deduzca una aproximación para $f'(x)$ que tenga un error de la forma $\sum_{i=1}^{\infty} C_i h^i$, donde C_i es una constante que no depende de h. Es decir, halle una expresión T_h tal que: $f'(x) = T_h + \sum_{i=1}^{\infty} C_i h^i$.

   b) Deduzca una aproximación para $f'(x)$ que tenga un error de la forma $\sum_{i=2}^{\infty} C_i h^i$, donde C_i es una constante que no depende de h. Es decir, halle una expresión T_h tal que: $f'(x) = T_h + \sum_{i=2}^{\infty} C_i h^i$.

   c) Usando la misma idea de los incisos anteriores, ¿puede hallar una expresión para $f''(x)$? ¿Cuál es la expresión del error?

6. Para calcular la derivada de una función $f(x)$ en el punto x_0, se puede usar la siguiente aproximación $f'(x) \approx T_h = \frac{f(x+h)-f(x)}{h}$. En general esta aproximación no es muy buena para valores "grandes" de h. Disminuir el tamaño de h no mejora "notablemente"[1] la aproximación. Sin embargo, las siguientes operaciones sí mejoran notablemente la aproximación:

   1. Calcular T_h para un h determinado.
   2. Calcular $T_{\frac{h}{2}}$ para el mismo h del inciso anterior.

      ?

   3. Calcular $T = 2 * T_{\frac{h}{2}} - T_h$.

   El valor obtenido en T es "notablemente mejor" que los calculados en T_h y $T_{h/2}$.

   a) La afirmación anterior estóa difícil de creer, así que verifíquela computacionalmente, utilizando varias funciones y varios puntos.

[1] La mejoría es solo lineal. ¿Por qué?

b) Una vez que haya visto los resultados del inciso anterior, demuestre la afirmación.[2]

c) Una vez demostrado el inciso anterior, explique qué significa en este contexto la frase "notablemente mejor".

d) Explique por qué no se obtienen mejores resultados si se realizan las mismas operaciones con la aproximación

$$T_h = \frac{f(x+h) - f(x-h)}{2h}.$$

e) Diga qué modificaciones habría que hacerle a los pasos 1., 2. y 3. para que la aproximación definida en el inciso anterior sí "mejore notablemente".

7. Una forma de aproximar la integral de una función $f(x)$ en un intervalo $[a, b]$ es interpolar la función en n puntos e integrar el polinomio de interpolación.

a) Diseñe un algoritmo que reciba un número n y devuelva una fórmula para hallar la integral de una función $f(x)$ en un intervalo $[a, b]$ utilizando un polinomio de interpolación con n puntos.

b) ¿Qué error se comete al aproximar la integral por la expresión propuesta por usted en el inciso anterior?

c) Utilice su algoritmo para hallar fórmulas de integración basada en 1,2, y 3 puntos, y los errores cometidos en cada caso.

8. Dados:

- $\mathbb{F}$ el conjunto de todas las funciones reales de variable real,
- $T : \mathcal{C} \subset \mathbb{F} \longrightarrow \mathbb{R}$, un operador que recibe como argumento una función real de variable real f y devuelve un número real[3], y
- T_h una aproximación del operador T que depende de un parámetro real h,[4]

Responda los siguientes incisos.

a) Asumiendo que $T = T_h + \sum_{j=i}^{\infty} C_j h^j$, demuestre que si T_h y $T_{\frac{h}{2}}$ son dos aproximaciones de T donde en una el valor de h es el doble que en la otra, entonces si se usa la aproximación:

$$T' = \frac{2^i T_{\frac{h}{2}} - T_h}{2^i - 1},$$

se obtiene un error con un orden $O(h^k)$, donde $k = i + 1$. O sea, se tiene que:

$$T = T' + \sum_{j=k}^{\infty} C_j h^j, \text{ con } k = i + 1.$$

[2]No es casual que en el ejericio anterior y en este aparezca la misma notación T_h.

[3]Como inciso extra de esta pregunta, mencione tres ejemplos de operadores de este tipo, y diga, para cada uno, cuál es su dominio $\mathcal{C}$.

[4]Otro inciso extra: para cada uno de los operadores definidos por usted en el inciso extra anterior, ponga ejemplos de aproximaciones T_h.

b) Tomando en cuenta que $f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h} + \sum_{i=1}^{\infty} C_i h^i$, diga cómo puede obtenerse una aproximación de $f'(x_0)$ que tenga un error de orden $O(h^n)$, para cualquier n, utilizando únicamente expresiones de la forma $T_h = \dfrac{f(x_0+h)-f(x_0)}{h}$.

c) Implemente, en el lenguaje de programación de su preferencia, un método que reciba una función $f : \mathbb{R} \longrightarrow \mathbb{R}$, un punto x_0, y un entero n, y devuelva una aproximación de $f'(x_0)$ que tenga un error de orden $O(h^n)$, utilizando la propuesta realizada por usted en el inciso anterior.

9. Dados:

   - $\mathbb{F}$ el conjunto de todas las funciones reales de variable real,
   - $T : \mathcal{C} \subset \mathbb{F} \longrightarrow \mathbb{R}$, un operador que recibe como argumento una función real de variable real f y devuelve un número real, y
   - T_h una aproximación del operador T que depende de un parámetro real h,

   responda los siguientes incisos.

   a) Asumiendo que $T = T_h + \sum_{j=i}^{\infty} C_{2j} h^{2j}$, y que T_h y $T_{\frac{h}{2}}$ son dos aproximaciones de T donde en una el valor de h es el doble que en la otra:
   ¿Qué valores deben tener A y B para que la expresión: $T' = \frac{AT_{\frac{h}{2}} - T_h}{B}$, tenga un error con un orden $O(h^{2k})$, donde $k = i+1$?
   O sea, ¿qué valores tienen que tener A y B para que: $T = T' + \sum_{j=k}^{\infty} C_{2j} h^{2j}$, con $k = i+1$?

   b) Deduzca[5] un operador T_h tal que $f'(x_0) = T_h + \sum_{i=1}^{\infty} C_{2i} h^{2i}$.

   c) Dado un número n entero, diga cómo puede obtenerse una aproximación de $f'(x_0)$ que tenga un error de orden $O(h^{2n})$ utilizando únicamente el operador propuesto por usted en el inciso anterior, y extrapolación.

   d) Implemente, en el lenguaje de programación de su preferencia, un mótodo que reciba una función $f : \mathbb{R} \to \mathbb{R}$, un punto x_0, y un entero n, y devuelva una aproximación de $f'(x_0)$ que tenga un error de orden $O(h^{2n})$, utilizando la propuesta realizada por usted en el inciso anterior.

10. Calcule el valor aproximado de la integral de la función $f(x) = \frac{1}{x}$ en el intervalo $[a, b] = [1, 2]$, utilizando el método de Romberg hasta calcular las extrapolaciones posibles para $h = \frac{b-a}{4}$. ¿Cuál es el valor obtenido? Diga la estimación del error absoluto, y del error relativo si se conoce que el valor exacto es ln 2.

[5] O encuentre.

11. En la vida cotidiana se presentan ciertas magnitudes que tienen un comportamiento aleatorio, por ejemplo el nivel de colesterol en sangre de una persona elegida al azar de un cierto grupo clasificado por edades, o la estatura de un adulto seleccionado también aleatoriamente. Los valores de dichas magnitudes varían sobre un intervalo de números reales, pero podrían medirse o registrarse hasta un cierto valor, los estadísticos las llaman variables aleatorias continuas. Toda variable aleatoria continua X tiene una función de densidad de probabilidad f. Esto significa que la probabilidad de que X se encuentre entre a y b, se halla integrando f

$$P(a \leq X \leq b) = \int_a^b f(x)dx \tag{2.43}$$

En el caso de que el fenómeno aleatorio (calificaciones en las pruebas de aptitud, precipitación pluvial anual en un lugar dado, etc.) se modele por medio de una distribución normal, la función de densidad de probabilidad de la variable aleatoria X pertenece a la familia de funciones

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{(2\sigma^2)}} \tag{2.44}$$

Siendo μ la media y σ la desviación estándar, la cual mide cuan dispersos están los valores de X. Aplicando lo visto anteriormente se tiene que las calificaciones del cociente de inteligencia CI se distribuyen normalmente con media 100 y desviación estándar 15

- ¿Qué porcentaje de la población tiene una calificación CI entre 85 y 115?
- ¿Qué porcentaje tiene por encima de 140?

Para el cálculo de las integrales correr los programas tomando $\epsilon = 10^-5$,

- CSIMPR41 con N= 256 intervalos
- Su algoritmo de Romberg con 5 filas

Compare el número de subintervalos N y el paso final h con los cuales termina cada método. ¿Cómo influye la diferencia en los valores de N sobre el número total de evaluaciones de función requeridas? Comentar la importancia de esto desde el punto de vista computacional.

Nota: Una función es integrable en un dominio infinito o semi-infinito sólo si es significativamente distinta de cero en un dominio pequeño y tiende a cero cuando $x \longrightarrow \infty$. Elegir un valor adecuado de x tal que se cumpla esta condición para poder calcular la integral planteada.

12. Dada una función $f(x)$ cualquiera[6] se desea calcular $I(f) = \int_0^\infty f(x)dx$.

    a) Diga cómo se puede calcular eficientemente el valor $I(f)$ para "cualquier" función $f(x)$.

    b) ¿Qué error se comete al utilizar su propuesta?

13. Dada una función $f(x)$ cualquiera se puede aproximar la integral $\int_a^b f(x)dx$ como una suma: $w_1 f(x_1) + w_2 f(x_2) + \cdots + w_n f(x_n)$, donde w_i son números reales seleccionados convenientemente.

[6]En realidad $f(x)$ no puede ser una función cualquiera. ¿Qué tiene que cumplir $f(x)$ para que se pueda utilizar en este ejercicio?

a) Diga qué valores deben tener los números w_i, de forma que las integrales $\int_a^b f_i(x)dx$ se puedan calcular de manera exacta para todas las funciones $f_i(x) = x^i$, con $i = \overline{0, n}$.

b) Diseñe e implemente un algoritmo que reciba el valor de n y devuelva los valores de las constantes w_i, con $i = \overline{1, n}$.

Bibliografía recomendada

- *Numerical Methods. 9th Edition.* J. D. Faires y R. L. Burden. Brooks Cole Publishing, 2011.
- *Elementary Numerical Analysis, An algorithmic approach. 3rd Edition.* S. D. Conte y Carl de Boor. McGraw-Hill Book Company. 1980.
- *Numerical Mathematics. Texts in applied mathematics;37.* A. Quarteroni, R. Sacco, F. Saleri. Springer Verlag, 2000.
- *Numerical Computing with Matlab.* C. Moler. 2004.
- *Numerical Analysis. Second* Edition. Walter Gautschi. Birkhauser 2012.

Capítulo 3

Ecuaciones diferenciales ordinarias

Introducción

Una ecuación diferencial ordinaria (EDO) de orden n es una expresión de la forma:

$$F(x, y, y', ..., y^{(n)}) = 0 \qquad ó \qquad y^{(n)} = f(x, y, y', ..., y^{(n-1)}), \tag{3.1}$$

que incluye una función $y(x)$ desconocida, su derivada n-ésima $y^{(n)}$ y algunas de las derivadas y' y'' hasta $y^{(n-1)}$. En este curso se consideran modelos dados por la segunda expresión de (3.1), donde la derivada de mayor orden $y^{(n)}$ viene dada de forma explícita. Específicamente, se considera el problema de valores iniciales o problema de Cauchy:

$$\begin{aligned} y^{(n)} &= f(x, y, y', ..., y^{(n-1)}), \quad x \in [a, b] \\ & y(a), y'(a), ..., y^{(n-1)}(a) \end{aligned} \tag{3.2}$$

La necesidad de usar los métodos numéricos surge debido a que la mayoría de las ecuaciones diferenciales que se presentan en la práctica no son lineales, o si lo son, no tienen coeficientes constantes, y algunas veces los coeficientes son empíricos. Para estos casos, no se cuenta con métodos exactos de solución.

Teniendo en cuenta que una EDO de orden n puede ser transformada en un sistema de n ecuaciones diferenciales ordinarias de primer orden, no se necesitan métodos especiales para resolverlas; una extensión de los métodos para resolver una EDO de primer orden a la solución de sistemas, es suficiente.

3.1. Problema de valores iniciales para una ecuación diferencial ordinaria de primer orden

Un problema de valores iniciales genérico para una ecuación diferencial ordinaria de primer orden se puede formular como: Encontrar una función $y : I \to D$ que satisface

$$y' = f(x, y) \tag{3.3}$$

$$y(x_0) = y_0 \tag{3.4}$$

donde $I \subset \mathbb{R}, D \subset \mathbb{R}^d$ y la parte derecha $f : I \times D \to \mathbb{R}^d$ es una función continua. Si $d > 1$, se tiene un sistema de ecuaciones diferenciales ordinarias de primer orden

$$y_1' = f_1(x, y_1, ..., y_d) \tag{3.5}$$
$$y_2' = f_2(x, y_1, ..., y_d) \tag{3.6}$$
$$\dots \tag{3.7}$$
$$y_d' = f_d(x, y_1, ..., y_d) \tag{3.8}$$
$$\tag{3.9}$$

La solución de una ecuación diferencial con condición inicial (3.4), con parte derecha continua, es una función $y : I \to D$ con derivada continua con respecto a x, que satisface la ecuación diferencial y la condición inicial en (3.4).

Entonces, aproximar numéricamente la solución del problema de valores iniciales (PVI) (3.4) significa:

- Dados un valor inicial x_0, un estado inicial y_0 y un valor final $\overline{X}$, calcular $y(\overline{X})$, donde la función $x \to y(x)$ es solución de (3.4).
- Se asume que el intervalo de interés $[x_0, \overline{X}]$ está dentro del dominio de definición de la función y.
- Como salida del método numérico que se emplee se obtiene un vector de n valores discretos $y_i \approx y(x_i)$, $i = 1, ..., n$. Los x_i determinan una partición equidistante del intervalo $[x_0, \overline{X}]$, $x_i = x_o + ih$

Existen diferentes métodos para aproximar numéricamente la solución de (3.4),

- los de paso simple, o de un solo paso, que se obtienen a partir del desarrollo en serie de Taylor de la solución $y(x+h)$ en términos de $y(x)$. En estos el valor aproximado y_{i+1} depende sólo del valor aproximado anterior y_i,. Los de Runge-Kutta son métodos de paso simple. El método de Euler es un método de Runge-Kutta de primer orden.
- los de paso múltiple (*multistep*) o de varios pasos, que se obtienen de integrar la ecuación diferencial, y donde el valor aproximado y_{i+1} depende de $k+1$ valores anteriores $y_i, y_{i-1,...,}y_{i-k}$.. Estos comprenden los métodos de Adams y el esquema predictor-corrector.

Tanto los métodos de paso simple como los métodos múltiple pueden ser implícitos o explícitos.

Lo más sencillo para resolver de forma aproximada el PVI (3.4) es aplicar uno de los esquemas más simples vistos para la aproximación numérica de la primera derivada. Por ejemplo tomando la aproximación hacia delante de la primera derivada en el punto $x = x_0$,

$$\begin{aligned} y'(x_0) \approx & \frac{y(x_0 + h) - y(x_0)}{h} \\ \approx & f(x_0, y(x_0)) \end{aligned}$$

de ahí que

$$y(x_0 + h) \approx y(x_0) + hf(x_0, y(x_0)) \tag{3.10}$$

La expresión (3.10) es lo que más adelante formalizaremos como método de Euler explícito. Si aproximamos la derivada con el modelo en diferencias hacia atrás, es decir

$$\begin{aligned} y'(x_0) \approx & \frac{y(x_0) - y(x_0 - h)}{h} \\ \approx & f(x_0, y(x_0)) \end{aligned}$$

entonces

$$y(x_0) \approx y(x_0 - h) + hf(x_0, y(x_0)). \tag{3.11}$$

Como se observa la función de la parte derecha se está evaluando en el punto donde se está aproximando la solución; esta expresión define lo que se conoce como método de Euler implícito.

Como se vio en la sección de diferenciación numérica la manipulación matemática del desarrollo en serie de Taylor nos permite obtener aproximaciones de distintos órdenes para las derivadas. Haciendo uso de este resultado es que se irán formalizando los métodos de paso simple.

Aunque nosotros presentaremos primero los métodos de paso simple llamados métodos de Runge-Kutta de diferentes órdenes y luego los métodos de paso múltiple, este no es exactamente el orden cronológico en que aparecieron los métodos, veamos la idea. El método de Euler se puede obtener también mediante un proceso de integración de la ecuación diferencial, considerando que se conoce el valor inicial $y(x_0)$ y por tanto se puede evaluar $f(x_0, y_0)$, resulta

$$\begin{aligned} \int_{x_0}^{x} y'(x)dx = & \int_{x_0}^{x} f(x, y)dx \\ y(x) - y(x_0) = & (x - x_0)f(x_0, y_0) \\ y(x) = & y(x_0) + (x - x_0)f(x_0, y_0) \end{aligned}$$

Haciendo $x = x_0 + h$, $y(x_0 + h) = y(x_0) + (h)f(x_0, y_0)$ se obtiene la fórmula de Euler. Si en lugar de aproximar la función $f(x, y)$ por una constante (polinomio de grado cero), la aproximamos por un polinomio de mayor orden $p_n(x)$ pues es de esperar que se obtenga una fórmula de un mayor orden de aproximación. Sin embargo para aproximar $f(x, y)$ por un polinomio de grado n se necesita conocer la solución y en $n + 1$ puntos y esta fue la idea de Adams y Moulton [1] que los llevó a desarrollar los métodos de paso múltiple que veremos más adelante.

3.1.1. Integración por serie de Taylor

Retomando la idea de obtener una aproximación para la primera derivada de un orden mayor a partir de tomar más términos en el desarrollo en serie de Taylor, aparecieron los llamados métodos de integración por serie de Taylor. Dada la ecuación diferencial $y' = f(x, y)$ donde la función f es suficientemente diferenciable con respecto a x y a y. Si $y(x)$ es la solución exacta del problema de Cauchy (3.4),

$$y' = f(x, y),\ y(x_o) = y_o,$$

[1] Los métodos de paso múltiple fueron presentados por primera vez por John Couch Adams para resolver un problema de F. Bashforth, alrededor de 1855, aunque fueron publicados in Bashforth 1883

$y(x)$ se puede desarrollar en serie de Taylor en un entorno del punto x_o :

$$y(x) = y(x_o) + (x - x_o)\ y'(x_o) + \frac{(x - x_o)^2}{2}\ y''(x_o) + \cdots$$

Las derivadas que aparecen en este desarrollo no se conocen explícitamente, puesto que la solución $y(x)$ tampoco se conoce; sin embargo, si f es suficientemente diferenciable, éstas pueden obtenerse derivando sucesivamente con respecto a x, teniendo en cuenta que y depende también de x :

$$y' = f \quad \prec \ \overset{x}{\underset{y}{\longrightarrow}} x$$

Así, para las primeras derivadas obtenemos:

$$\begin{aligned} y' &= f(x, y) \\ y'' &= \frac{\partial f}{\partial x} + f\ \frac{\partial f}{\partial y} \\ y''' &= \frac{\partial^2 f}{\partial x^2} + 2f\frac{\partial^2 f}{\partial x \partial y} + \frac{\partial^2 f}{\partial y^2}\ f^{\,2} + (\frac{\partial f}{\partial y})^2 f + \frac{\partial f}{\partial x}\ \frac{\partial f}{\partial y} \\ &\vdots \end{aligned}$$

Como se observa la complejidad de este método aumenta a medida que aumenta el orden de la derivada que se calcula y depende también de la complejidad de la función que aparece en la parte derecha de la ecuación diferencial, no obstante veamos cómo funciona.

Ejemplo 36 *Usando el desarrollo en serie de Taylor, calcular los valores de* $y(1/5)$ *y* $y(2/5)$ *para el problema de Cauchy*

$$y' = 1 + xy + y^2,\ y(0) = 0,$$

Derivando obtenemos,

$$\begin{aligned} y'' &= y + xy' + 2yy' \\ y''' &= 2y'(1 + y') + y''(x + 2y) \\ &\vdots \end{aligned}$$

y evaluando en $x_o = 0$,

$$y'(0) = 1,\ y''(0) = 0,\ y'''(0) = 4$$

Luego,

$$\begin{aligned} y(x) &= y_o + (x - x_o)y'_o + \frac{(x - x_o)^2}{2}y''_o + \frac{(x - x_o)^3}{3!}y'''_o + \cdots \\ &= 0 + (x - 0)1 + 0 + \frac{(x - 0)^3}{6}4 + \cdots \\ &= x + \frac{2}{3}x^3 + \cdots \end{aligned}$$

Truncando en el término de la tercera derivada, y evaluando en $x = x_o + h = 0 + \frac{1}{5} = 0{,}2$, *el resultado del primer paso será:*

$$y(\frac{1}{5}) \approx \frac{1}{5} + \frac{2}{3}(\frac{1}{5})^3 = 0{,}2054$$

Si ahora se considera $x_o = \frac{1}{5}$ con condición inicial $y(\frac{1}{5})$, y se aplica nuevamente el desarrollo en serie de Taylor con $x = x_1 = \frac{1}{5} + h = \frac{2}{5} = 0{,}4$, se obtiene el resultado de un segundo paso:

$$y(\frac{2}{5}) \approx y(\frac{1}{5}) + \frac{1}{5}y'(\frac{1}{5}) + \frac{(1/5)^2}{2}y''(\frac{1}{5}) + \frac{(1/5)^3}{6}y'''(\frac{1}{5}) = 0{,}4461$$

lo que podría repetirse hasta obtener $y(x_n) \approx y(x_o+nh)$. Utilizando así la serie de Taylor, se obtiene un método aproximado del tipo paso-por-paso.

Está claro que, a menos que $f(x, y)$ sea una función muy sencilla como en este caso, las derivadas superiores se van haciendo progresivamente más complejas. Por razones prácticas debe limitarse el número de términos, lo que implica una limitación en la precisión de la solución aproximada $y_h(x_{i+1})$.

El **error de método o error de truncamiento local** de $y_h(x_{i+1})$ está dado por el resto de la serie

$$E_r = \frac{h^{r+1}}{(r+1)!}y^{(r+1)}(\xi, y(\xi)), \quad x_i < \xi < x_i + h,$$

si para el cálculo de y_h se trunca en el término de orden r. Se considera entonces la serie de Taylor truncada como algoritmo de Taylor con precisión de orden r y error de truncamiento local $\Theta(h^{r+1})$.

3.1.2. Método de Euler

El método de Euler[2] es el más sencillo de los métodos de paso simple para resolver el problema de Cauchy de primer orden y como se vio una de las vías de deducirlo es truncando el desarrollo en serie de Taylor en el término de la primera derivada. Se utiliza para demostrar la existencia de la solución de dicho problema, así como para resolverlo numéricamente. Se considera una partición del intervalo de integración

$$x_0, x_1, ..., x_{n-1}, x_n = X \tag{3.12}$$

y se reemplaza en cada subintervalo la solución por el primer término de la serie de Taylor

$$\begin{aligned} y_1 - y_0 &= (x_1 - x_0)\, f\,(x_0, y_0) \\ y_2 - y_1 &= (x_2 - x_1)\, f\,(x_1, y_1) \\ &\vdots \\ y_n - y_{n-1} &= (x_n - x_{n-1})\, f\,(x_{n-1}, y_{n-1}) \end{aligned} \tag{3.13}$$

denotando a $h = (h_0, h_1, .., h_{n-1})$, $h_i = x_{i+1} - x_i$. Conectando $y_0, y_1,\ y_1, y_2, ..etc.$ por líneas rectas obtenemos el polígono de Euler

$$y_h = y_i + (x - x_i)\ \ f(x_i, y_i),\ \ x_i \le x \le x_{i+1}$$

Analíticamente el método de Euler se obtiene al truncar el desarrollo en serie de Taylor en un entorno del punto $a = x_o$,

$$y(x_o + h) = y(x_o) + h\ y'(x_o) + \frac{h^2}{2!}\ y''(x_o) + \cdots,$$

[2]Leonhard Euler (15-4-1707, Suiza,-18-9-1783, Rusia), explicó este método en 1768 en la última sección de su Institutiones Calculi Integralis

después del término con la derivada de primer orden

$$y_1 = y_h(x_o + h) = y(x_o) + h\ f(x_o, y_o).$$

Si de forma análoga a partir del punto (x_i, y_i) hallamos $y(x_i + h) = y_{i+1}$, se obtiene la expresión general de la fórmula de Euler:

$$y_{i+1} = y_i + h\ f(x_i, y_i),\ i = 0, 1, 2, \cdots \tag{3.14}$$

con error de método o **error de truncamiento local**

$$E = \frac{h^2}{2}\ y''(\xi) = \Theta(h^2), \quad x_i < \xi < x_{i+1}.$$

Sin embargo más que nada nos interesa el error que se comete al considerar la solución aproximada $y_h(x_i)$ como aproximación de la solución exacta $y(x_i)$. Este error se ve afectado por el error que se comete en cada paso conocido como **error de discretización local**, así como por la estabilidad del algoritmo; aspectos que serán abordados más adelante.

Ejemplo Aplicando el método de Euler al ejemplo anterior, $y' = 1 + xy + y^2$, $y(0) = 0$, con igual paso de integración h=0.2, obtenemos:
$y_o = 0$
$y_1 = 0 + h\ f(0, 0) = 0 + 0{,}2(1) = 0{,}2$
$y_2 = 0{,}2 + h\ f(0{,}2, 0{,}2) = 0{,}2 + 0{,}2(1 + (0{,}2)(0{,}2) + (0{,}2)^2) = 0{,}416$,

3.1.3. Error de discretización local

Antes de definir formalmente el error de discretización local, veamos intuitivamente cómo interpretar los términos que le dan nombre. Error local significa el error que se comete en cada paso al aproximar la solución exacta en el tiempo x_{i+1} por la fórmula dada, asumiendo que en el tiempo anterior x_i la solución es exacta, y de discretización porque como se verá este error mide cuán bien la solución exacta satisface el modelo discreto, veamos.

Para los métodos de un sólo paso se tiene

$$y_{i+1} = y_i + h\ \Phi(x_i, y_i, h_i),\ i = 0, 1, 2, \cdots \tag{3.15}$$

donde $\Phi(x_i, y_i, h_i)$ es llamada función de método. Si denotamos la solución exacta en el tiempo x_{i+1} como $y(x_{i+1})$ y asumimos se conoce $y(x_i)$, entonces

$$\begin{aligned} \frac{y(x_{i+1}) - y_{i+1}}{h} =& \frac{y(x_{i+1}) - (y_i + h\ \Phi(x_i, y_i, h_i))}{h} \\ =& \frac{y(x_{i+1}) - y(x_i)}{h} - \Phi(x_i, y(x_i), h_i) \\ =& L(x_i, h) \end{aligned} \tag{3.16}$$

A la diferencia anterior es a lo que se conoce como error de discretización local en el punto $(x_i, y(x_i))$ y en la mayoría de los textos se denota por $L(x_i, h)$. Hay autores que toman el error de discretización

local como el mayor de todos los errores locales calculados paso a paso y lo denotan por $L(h) = \underbrace{max}_{a \leq x_i \leq b-h} |L(x_i, h)|$. Como se observa la expresión (3.16) es una medida de cuán bien la solución exacta satisface el método de un solo paso dado.

En el método de Euler $\Phi(x_i, y(x_i), h_i) = f(x_i, y(x_i))$ y se tiene que el error de discretización local $L(h) = O(h)$.

El error de discretización local está relacionado con el concepto de consistencia del método numérico. Asumiendo una notación diferente para el error de discretización local $\delta_{i+1}(x_i, y_h^i, h)$, con

$$\delta_{i+1}(x_i, y_h^i, h) = \Delta(x_i, y_h^i, h) - \Phi(x_i, y_i, h) \tag{3.17}$$

donde $\Delta(x_i, y_h^i, h) = \frac{y(x_{i+1}) - y(x_i)}{h}$.

Definición 37 *Un método de un solo paso se dice consistente con el problema de valores iniciales o consistente de orden p si*

$$max_{x_i \in [a,b]} \|\delta_{i+1}\| = 0 \tag{3.18}$$

o

$$max_{x_i \in [a,b]} \|\delta_{i+1}\| = O(h^p) \tag{3.19}$$

respectivamente.

El método de Euler es entonces consistente de orden uno.

3.1.4. Error global y estabilidad en el método de Euler

El error global está relacionado con el análisis de estabilidad, veamos.

Considerando el algoritmo de Taylor para $r = 1$ y denotando el valor exacto de la solución en x_i por $y(x_i)$:

$$y(x_{i+1}) = y(x_i) + h\ y'(x_i) + \frac{h^2}{2}\ y''(\xi), \quad x_i < \xi < x_{i+1},$$

y por y_i el valor aproximado, calculado por la fórmula de Euler:

$$y_{i+1} = y_i + h\ f(x_i, y_i),$$

entonces el error global en x_{i+1} se calcula como

$$y(x_{i+1}) - y_{i+1} = \{y(x_i) - y_i + h\ [f(x_i, y(x_i)) - f(x_i, y_i)]\} + \frac{h^2}{2}\ y''(\xi). \tag{3.20}$$

En (3.20) la parte del miembro derecho entre llaves es cero en el primer paso pues $y(x_o) = y_o$ lo que implica que $f(x_o, y(x_o)) = f(x_o, y_o)$, y el resto $\frac{h^2}{2} y''(\xi)$ es el error de truncamiento.

En general la expresión entre llaves de (3.20) representa el error propagado (EP) y aplicando el teorema del valor medio para la variable y se obtiene

$$\begin{aligned} EP &= y(x_i) - y_i + h\ [\frac{\partial f}{\partial y}(x_i, \eta)(y(x_i) - y_i)] \\ &= (y(x_i) - y_i)\ (1 + h\ J_i), \end{aligned} \tag{3.21}$$

donde J_i denota la derivada parcial en (x_i, η), y $(1 + h\ J_i)$, el factor de propagación. Sustituyendo (3.21) en (3.20), denotando el error global y el error de truncamiento local por EG y ETL respectivamente, se obtiene:

$$EG_{i+1} = EG_i(1 + h\ J_i) + ETL_{i+1}.$$

De la expresión anterior se deduce que para que no crezca el error de un paso a otro se debe cumplir que $|1 + h\ J_i| < 1$, desigualdad que nos da una restricción para el paso de discretización. Sin embargo como la solución exacta no se conoce pues lo que se hace es analizar **¿cómo varía** $y_h(x)$ **cuando el valor inicial** y_0 **cambia?.**

Sea z_0 otro valor inicial $y(x_0) = z_0$ (la partición es la misma). Calculemos

$$z_1 - z_0 = (x_1 - x_0) f(x_0, z_0) \tag{3.22}$$

Necesitamos estimar $|z_1 - y_1|$. Restando (3.22) de la primera línea de (3.13)

$$\begin{aligned} y_1 - y_0 - z_1 + z_0 &= (x_1 - x_0) f(x_0, y_0) - (x_1 - x_0) f(x_0, z_0) \\ z_1 - y_1 &= -y_0 + z_0 + (x_1 - x_0)\left[f(x_0, z_0) - f(x_0, y_0)\right]. \end{aligned}$$

Entonces necesitamos estimar $[f(x_0, z_0) - f(x_0, y_0)]$ y aplicando el teorema del valor medio tendremos

$$[f(x_0, z_0) - f(x_0, y_0)] = \frac{\partial f}{\partial y}(x_0, \eta_0)(z_0 - y_0), \quad z_0 < \eta_0 < y_0$$

. De aquí que

$$z_1 - y_1 = (z_0 - y_0)\left\{1 + h\frac{\partial f}{\partial y}(x_0, \eta_0)\right\} \tag{3.23}$$

Si comparamos (3.23) con (3.21) obtenemos que la estabilidad del método de Euler dependerá de si el factor de propagación $(1 + hJ_i)$ representa una amplificación o una reducción del error en cada paso i. Como el factor de propagación depende del paso de integración h, es manejable y se puede lograr una reducción del mismo exigiendo que $|1 + h\ J_i| < 1$, es decir, que

$$\begin{aligned} -1 < 1 + h\ J_i < 1 \\ 0 < 2 + h\ J_i < 2, \end{aligned}$$

y como $h > 0$, debe ser $J_i < 0$, luego para la estabilidad numérica del método de Euler debe cumplirse en cada paso que

$$-\frac{2}{J_i} > h > 0.$$

Pero aún garantizándose paso a paso la estabilidad numérica, hay acumulación de errores, y como demostramos el error de discretización local es una $O(h)$, de ahí que el error global sea de orden h. ¿Cómo obtener entonces más precisión en la solución numérica sin disminuir h demasiado que aumente el número de pasos al aplicar el método de Euler y con ello el error global, y sin usar muchos términos del desarrollo en serie de Taylor que aumenten la complejidad del cálculo excesivamente por la necesidad de obtener y evaluar las derivadas totales sucesivas ?

La respuesta a esta pregunta está dada por los métodos de Runge-Kutta que se presentan a continuación.

3.2. Los métodos de Runge-Kutta

La idea de los métodos de Runge-Kutta [3], pensados y descritos por Runge en 1895 y elaborados más ampliamente por su colaborador Kutta en 1901 consiste en calcular la nueva ordenada y_{i+1} adicionando a la anterior y_i un incremento Δy_i que coincida con el desarrollo de Taylor de $y(x_i+h)$ hasta el término de la derivada de orden r, pero que sólo use la primera derivada f, sin requerir la evaluación de derivadas superiores. Este incremento Δy_i se obtiene como combinación lineal de valores de $y' = f$. Estos valores corresponden a la evaluación de f en r puntos del subintervalo $[x_i, x_i + h]$:

$$\begin{aligned} y_{i+1} &= y_i + \Delta y_i \\ &= y_i + \sum_{m=1}^{r} b_m k_m \end{aligned} \tag{3.24}$$

donde

$$k_m = h\ f(\xi_m, \eta_m)$$

$$\xi_m = x_i + c_m h, \quad c_1 = 0 \ \text{ por definición, y } 0 < c_m \leq 1 \text{ para } m > 1$$

$$\eta_m = y_i + a_{m1}k_1 + a_{m2}k_2 + \cdots + a_{m,m-1}k_{m-1} = y_i + \sum_{j=1}^{m-1} a_{mj}k_j.$$

Los parámetros c_m, a_{mj} y b_m que aparecen en las expresiones de ξ_m, η_m y Δy_i se determinan bajo la condición de que el valor aproximado y_{i+1} calculado según (3.24) coincida con el que se obtendría evaluando el desarrollo en serie de Taylor hasta el término de orden r :

$$y_{i+1} = y_i + \sum_{m=1}^{r} \frac{h^m}{m!} y_i^{(m)} \tag{3.25}$$

lo cual equivale a exigir que el incremento de Runge coincida con el incremento de Taylor:

$$\sum_{m=1}^{r} b_m k_m = \sum_{m=1}^{r} \frac{h^m}{m!} y_i^{(m)}. \tag{3.26}$$

En el miembro izquierdo de (3.26) se observa que el incremento se construye como combinación lineal de las funciones k_m, es decir, de la función f evaluada en r puntos con abscisa $\xi_m \in [x_i, x_i + h]$, mientras que en la segunda, el incremento se construye como combinación lineal de las r primeras derivadas de y, evaluadas en $x = x_i$. Es decir, la idea fundamental de los métodos de Runge-Kutta consiste en contraponer dos formas de construir el incremento Δy_i :

$$\left[\begin{array}{c} Taylor: \\ \text{Evaluación de } r \text{ funciones (las} \\ \text{derivadas) en 1 solo punto } x_i \end{array}\right] vs. \left[\begin{array}{c} Runge-Kutta: \\ \text{Evaluación de 1 función } (f) \\ \text{en } r \text{ puntos del intervalo } [x_i, x_{i+1}] \end{array}\right]$$

Desde el punto de vista computacional, es más eficiente evaluar una sola función en r puntos, que hallar las derivadas superiores de y' y evaluarlas todas en el punto x_i, como se requeriría en el desarrollo en serie de Taylor. De ahí la vigencia de los métodos de Runge-Kutta un siglo después de ser propuestos.

[3] Martin Wilhelm Kutta (3 de Noviembre de 1867 en Pitschen, Silecia norte, actual Byczyna, Polonia- 25 de Diciembre de 1944 en Furstenfeldbruck, Alemania). Carle David Tolmé Runge, (30 de Agosto de 1856 en Bremen, Alemania- 3 de Enero de 1927 en Goettingen Alemania).

3.2.1. Deducción de las fórmulas de segundo orden

Siendo $r = 2$, el incremento estará definido por

$$\Delta y_i = b_1 k_1 + b_2 k_2$$

donde

$$\begin{aligned} k_1 &= h\ f(\xi_1, \eta_1) = h\ f(x_i, y_i) \\ k_2 &= h\ f(\xi_2, \eta_2) = h\ f(x_i + c_2 h, y_i + a_{21} k_1). \end{aligned}$$

Los coeficientes desconocidos se determinan de manera que

$$y_{i+1} = y_i + b_1 k_1 + b_2 k_2 \tag{3.27}$$

coincida con el desarrollo de Taylor en el punto x_i hasta el término de segundo orden:

$$y_{i+1} = y_i + h y_i' + \frac{h^2}{2} y_i''. \tag{3.28}$$

Sustituyendo $y' = f(x, y),\ \ y'' = \frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y}$ en (3.28), obtenemos

$$y_{i+1} = y_i + h\ f(x_i, y_i) + \frac{h^2}{2}(\frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y})(x_i, y_i) \tag{3.29}$$

Por otra parte, usando el desarrollo en serie de Taylor para la función de dos variables $k_2(c_2, a_{21})$ obtenemos

$$\begin{aligned} k_2(x_i + c_2 h, y_i + a_{21} k_1) &= h\ f(x_i + c_2 h, y_i + a_{21} k_1) \\ &= h[f(x_i, y_i) + (c_2 h \frac{\partial}{\partial x} + a_{21} k_1 \frac{\partial}{\partial y})\ f(x_i, y_i) \\ &+ \frac{1}{2!}(c_2 h \frac{\partial}{\partial x} + a_{21} k_1 \frac{\partial}{\partial y})^2\ f(x_i, y_i) + \cdots \end{aligned}$$

Sustituyendo k_1 por $hf(x_i, y_i)$ y k_2 por la expresión anterior en (3.27), obtenemos después de agrupar según potencias de h :

$$y_{i+1} = y_i + h(b_1 + b_2)\ f(x_i, y_i) + h^2[b_2(c_2 \frac{\partial f}{\partial x} + a_{21} f \frac{\partial f}{\partial y})(x_i, y_i)] + O(h^3). \tag{3.30}$$

Igualando (3.29) y (3.30) término a término, obtenemos para los coeficientes de h

$$(b_1 + b_2) f(x_i, y_i) = f(x_i, y_i) \Longrightarrow b_1 + b_2 = 1,$$

y para los coeficientes de h^2

$$b_2(c_2 \frac{\partial f}{\partial x} + a_{21} f \frac{\partial f}{\partial y})(x_i, y_i) = \frac{1}{2}(\frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y})(x_i, y_i) \Longrightarrow b_2 c_2 = \frac{1}{2},\ \ b_2 a_{21} = \frac{1}{2},$$

es decir, tenemos 3 ecuaciones para la determinación de las 4 incógnitas b_1, b_2, c_2, a_{21}, luego existe un grado de libertad.

Tomando b_2 como una constante arbitraria, $b_2 = \theta$ tal que $0 < c_2 \leq 1$, según las hipótesis de los métodos de Runge-Kutta tenemos que $c_2 = \frac{1}{2\theta} \leq 1$, o sea, $\theta \geq \frac{1}{2}$. Entonces queda

$$b_2 = \theta,\ b_1 = 1 - \theta,\ c_2 = a_{21} = \frac{1}{2\theta},\ \text{con}\ \ \theta \geq \frac{1}{2}. \tag{3.31}$$

Estos coeficientes que intervienen en la formulación de los métodos de Runge-Kutta se encuentran agrupados en el conocido como arreglo de Butcher [4] que para $r = 2$ se ordena como sigue:

$$\begin{array}{c|cc} c_1 & 0 & 0 \\ c_2 & a_{21} & 0 \\ \hline & b_1 & b_2 \end{array}$$

Casos particulares

- $\theta = 1 \implies b_2 = 1\ b_1 = 0\ c_2 = a_{21} = \frac{1}{2}$

$$y_{i+1} = y_i + k_2,$$
$$k_2 = hf(x_i + \frac{h}{2}, y_i + \frac{h}{2}f(x_i, y_i))$$

  Fórmula de orden 2 basada en la regla de punto medio, se conoce como método de Euler modificado

- $\theta = \frac{1}{2} \implies b_2 = \frac{1}{2},\ b_1 = \frac{1}{2},\ c_2 = a_{21} = 1$:

$$y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2),$$
$$k_1 = hf(x_i, y_i),\ k_2 = hf(x_i + h, y_i + k_1)$$

  Fórmula de orden 2 basada en la regla de los trapecios y conocida como método de Heun

Para otros valores admisibles de θ se obtienen otras fórmulas de Runge-Kutta de 2do. orden, todas con error de método $O(h^3)$.

3.2.2. Fórmulas de orden superior

De manera análoga se pueden deducir fórmulas de orden r, $r > 2$, es decir, que logran una coincidencia de la solución aproximada dada por el método con $r > 2$ términos del desarrollo de Taylor, con lo cual se logra mayor precisión. Citemos, como ejemplo, la fórmula de Runge-Kutta de 4to. orden, que es una de las más usadas

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{3.32}$$

[4] J.C. Butcher, Numerical Methods for Ordinary Differential Equations, Second Edition, 2008 John Wiley and Sons Ltd.

donde

$$
\begin{aligned}
k_1 &= h\ f(x_i, y) \\
k_2 &= h\ f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1) \\
k_3 &= h\ f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2) \\
k_4 &= h\ f(x_i + h, y_i + k_3)
\end{aligned} \tag{3.33}
$$

con error de método $O(h^5)$, que se obtiene a partir de un sistema de 11 condiciones para la determinación de las 13 incógnitas b_1, b_2, b_3, b_4; c_2, c_3, c_4; $a_{21}, a_{31}, a_{32}, a_{41}, a_{42}, a_{43}$. Siguiendo el patrón de Butcher para las fórmulas de orden 4:

c_1	0	0	0	0
c_2	a_{21}	0	0	0
c_3	a_{31}	a_{32}	0	0
c_4	a_{41}	a_{42}	a_{43}	0
	b_1	b_2	b_3	b_4

De manera general un arreglo de Butcher se tabula como sigue

c^T	A
	b

donde para un método de orden 4 c es un vector columna de 4 componentes, A es una matriz 4×4 estrictamente triangular inferior (estamos considerando los métodos explícitos), y b un vector fila de 4 componentes. Para la fórmula anterior se tiene:

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$

Esquema de cálculo

La aplicación de una fórmula de Runge-Kutta de orden r con $r \leqslant 4$ requiere exactamente r evaluaciones de la función $f(x, y)$ en cada paso. Para métodos de orden mayor que 4 se requieren en general más evaluaciones de función que el orden. Para organizar los cálculos en un método de orden r será conveniente en cada paso, el siguiente esquema:

i	x_i	y_i	$f(\xi,\eta)$	$h\ f$
0	x_o	y_o		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	x_i	y_i		Δy_{i-1}
-	$\xi_1 = x_i$	$\eta_1 = y_i$	$f(\xi_1,\eta_1)$	k_1
-	$\xi_2 = x_i + c_2 h$	$\eta_2 = y_i + a_{21}k_1$	$f(\xi_2,\eta_2)$	k_2
-	$\vdots$	$\vdots$	$\vdots$	$\vdots$
-	$\xi_r = x_i + c_r h$	$\eta_r = y_i + a_{r1}k_1 + \cdots + a_{r,r-1}k_{r-1}$	$f(\xi_r,\eta_r)$	k_r
$\cdots$	$\cdots$	$\Delta y_i = b_1k_1 + ... + b_rk_r$	$\cdots$	$\cdots$
i+1	$x_{i+1} = x_i + h$	$y_{i+1} = y_i + \Delta y_i$		Δy_i
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	$x_n = x_{n-1} + h$	$y_n = y_{n-1} + \Delta y_{n-1}$		Δy_{n-1}

Ejemplo: Dado el problema de Cauchy

$$y' = y\sin(x) + \cos(x) + 1, y(1) = 1,\ x \in [1,2]$$

Resolverlo usando la fórmula de Runge-Kutta de orden 2:

$$y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2),$$
$$k_1 = hf(x_i, y_i),\ k_2 = h\ f(x_i + h, y_i + k_1)$$

y aplicando el esquema del cálculo con paso $h = 0{,}1$.

Solución:

$$f(x,y) \equiv y\sin(x) + \cos(x) + 1$$
$$\xi_1 = x_i, \eta_1 = y_i,$$
$$\xi_2 = x_i + 0{,}1, \eta_2 = y_i + k_1,$$
$$\Delta y_i = \frac{1}{2}(k_1 + k_2)$$

i	x_i	y_i	$sin(x_i)$	$cos(x_i)$	$f(\xi,\eta)$	$h\ f$
0	1	1	—	—	—	—
-	$\xi_1 = 1$	$\eta_1 = 1$	,84147	,54030	2,38177	$k_1 =$,23818
-	$\xi_2 = 1{,}1$	$\eta_2 = 1{,}23818$	,89121	,45360	2,55708	$k_2 =$,25571
1	1.1	1.24695	—	—	—	$\Delta y_o =$,24695
-	$\xi_1 = 1{,}1$	$\eta_1 = 1{,}24695$	,89121	,45360	2,56489	$k_1 =$,25649
-	$\xi_2 = 1{,}2$	$\eta_2 = 1{,}50344$	,93204	,36236	2,76363	$k_2 =$,27636
2	1.2	1.51338	—	—	—	$\Delta y_1 =$,26643
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
10	2.0	$\cdots$	—	—	—	Δy_9

Algoritmo de Runge-Kutta con paso fijo

Dado el problema de Cauchy: $y' = f(x,y),\ \ x \in [a,b],\ \ y(a)$. Teniendo como parámetros fijos c, A, b para un método de orden r. Se toma como paso de integración fijo $h = (b-a)/n$, donde n es la cantidad de pasos y se desarrollan subrutinas auxiliares para evaluar $f(x,y)$

Algoritmo 38

Algoritmo de Runge-Kutta con paso fijo

Paso 1: leer c, A, b *y calcular* h
Paso 2: guardar a *en* x *y en* x_o
Paso 3: leer la condición inicial $y(a)$*, y guardarla en* y *y en* y_o
Paso 4: imprimir x, y
Paso 5: Para $i = 1 : n$*, repetir:*
5.1) poner $x \curvearrowright \xi$ *y* $\curvearrowright \eta$,
calcular $hf(\xi, \eta)$*, y guardarlo en* k_1
Para $m = 1 : r$
calcular $x + a_m h \curvearrowright \xi$
calcular $\sum_{j=1}^{m-1} a_{mj} k_j \curvearrowright z$, $y + z \curvearrowright \eta$
calcular $hf(\xi, \eta)$ *y guardarlo en* k_m
fin
5.2) resultados del nuevo paso:
calcular $x + h$ *y guardarlo en* x *y en* x_i
calcular $\sum_{j=1}^{r} p_{rj} k_j \curvearrowright z$, $y + z$*, y guardarlo en* y *y en* y_i
5.3) imprimir x, y
fin

3.2.3. Estimación del error

Aunque con las fórmulas de Runge-Kutta se sustituye el cálculo de las derivadas por evaluaciones de función en cada paso, para la estimación del error de método o de truncamiento en cada paso habría, sin embargo, que evaluar la derivada $y^{(r+1)}$, lo que exigiría previamente la obtención de las expresiones de las derivadas que antes se evitó hallar, y no tiene sentido.

De ahí que en la práctica se busquen vías alternativas más simples para la estimación del error. Un procedimiento es el llamado de doble cómputo, que fue usado para estimar el error en la integración numérica, y otra vía es usando simultáneamente dos fórmulas de paso simple de distinto orden, (por ejemplo, el algoritmo RKF45 que usa una fórmula de orden 4 y otra de orden 5).

3.2.4. Doble cómputo

Supongamos que estamos usando un método de Runge-Kutta con precisión local de orden r, y que llegamos al punto x_i con paso $h = x_i - x_{i-1}$. Se quiere ahora integrar desde x_i hasta $\overline{x} = x_{i+1} = x_i + h$ dos veces, la primera usando el paso actual h, y la segunda, usando dos pasos de longitud $h/2$. Se obtienen entonces dos estimaciones $y_h(\overline{x})$ y $y_{h/2}(\overline{x})$ del valor exacto de $y(\overline{x})$, y comparando estas estimaciones podemos obtener una estimación del error.

Partimos de que un método de Runge-Kutta de orden r tiene error de truncamiento $O(h^{r+1}) = \frac{h^{r+1}}{(r+1)!} y^{(r+1)}(\xi)$ en cada paso, y el error global en el punto $\overline{x} = x_i + mh$ se expresa de la forma

$$E_g(\overline{x}) = y(\overline{x}) - y_h(\overline{x}) = C(\overline{x})\ h^r + O(h^{r+1}). \tag{3.34}$$

Aquí, $y_h(\overline{x})$ denota el valor aproximado de la solución en el punto $\overline{x} = x_i + mh$ obtenido a partir de x_i, después de realizar m pasos de amplitud h con una cierta fórmula de Runge-Kutta, y la constante $C(\overline{x})$ depende de la función f y de $\overline{x}$, pero no de h. La expresión (3.34) no es computable, pues no se tiene $C(\overline{x})$. Nos proponemos eliminar la parte derecha de la misma para obtener la estimación del error en términos de valores computables. Aplicando (3.34) con $m = 1$, y paso h, es decir, efectuando un solo paso:

$$E_g = y(x_{i+1}) - y_h(x_{i+1}) = C(x_{i+1})h^r + O(h^{r+1}), \tag{3.35}$$

y con paso de amplitud $h/2$, efectuando dos cálculos

$$E_g = y(x_{i+1}) - y_{h/2}(x_{i+1}) = C(x_{i+1})(h/2)^r + O((h/2)^{r+1}). \tag{3.36}$$

Restando (3.36) de (3.35),

$$y_{h/2}(x_{i+1}) - y_h(x_{i+1}) \approx C(x_{i+1})h^r(1 - \frac{1}{2^r}), \tag{3.37}$$

de donde se puede despejar

$$C(x_{i+1})h^r \approx \frac{2^r(y_{h/2}(x_{i+1}) - y_h(x_{i+1})}{2^r - 1},$$

y sustituyendo en (3.35), obtener

$$E_g \approx \frac{2^r(y_{h/2}(x_{i+1}) - y_h(x_{i+1}))}{2^r - 1}, \tag{3.38}$$

que constituye la estimación del error global de la solución aproximada y_h en el punto $\overline{x}$. Esta estimación del error sí es computable, y no requiere la evaluación de C.

3.2.5. Dos fórmulas de distinto orden (RKF45)

Si en lugar de calcular dos aproximaciones de la solución en el mismo punto, usando dos tamaños de paso calculamos la aproximación de la solución y_{i+1} empleando dos fórmulas de Runge-Kutta con precisión de orden 5 y 6, es decir, cuyo error global es $O(h^4)$ y $O(h^5)$ respectivamente:

$$y_{i+1}^{(5)} = y_i^{(5)} + \sum_{j=1}^{5} b_{5j}k_j$$
$$y_{i+1}^{(6)} = y_i^{(6)} + \sum_{j=1}^{6} b_{6j}k_j,$$

el error global estará dado por

$$|E_g(\overline{x})| = \left| y_{i+1}^{(5)}(\overline{x}) - y_{i+1}^{(6)}(\overline{x}) \right|.$$

En estas dos fórmulas de Runge-Kutta, los c_m y los a_{mj} comunes son iguales, luego los k_j coinciden para $1 \leq j \leq 5$, mientras que los b_{5j} y b_{6j} difieren. La tabla de parámetros según el patrón de Butcher tendrá la forma:

$$\begin{array}{ccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_2 & a_{21} & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & 0 & 0 & 0 \\
c_5 & a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 \\
c_6 & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 \\
\hline
 & b_{51} & b_{52} & b_{53} & b_{54} & b_{55} & 0 \\
\hline
 & b_{61} & b_{62} & b_{63} & b_{64} & b_{65} & b_{66}
\end{array}$$

De aquí que aunque se usen dos fórmulas, el número de evaluaciones de función en cada paso es 6 ($k_1, \ldots, k_6$), luego el cálculo es más eficiente que por doble cómputo, el cual requeriría para una fórmula de Runge-Kutta de orden 4, 12 evaluaciones (4 con paso h y 8 más con paso $h/2$).

Aplicación al cambio de paso

Después de calcular $x_i + h \longrightarrow \overline{x}$, $y_i + \sum_{j=1}^{r} b_{rj}k_j \longrightarrow u$, v , bien sea por doble cómputo o por RKF45, entonces en el punto $\overline{x}$ se dispone de dos aproximaciones del valor de y :

$$u = y_h(\overline{x}) \ , \ v = y_{h/2}(\overline{x}) \quad \text{ó} \quad u = y^{(5)}(\overline{x}) \ , \ v = y^{(6)}(\overline{x})$$

respectivamente, con las cuales se puede estimar el error absoluto:

$$E(\overline{x}) \approx \frac{2^r(v-u)}{2^r-1} \quad \text{ó} \quad E(\overline{x}) \approx y^{(6)}(\overline{x}) \ - y^{(5)}(\overline{x}) = v - u.$$

Estrategia del cambio de paso

Dadas cotas ε_1 y ε_2 del error absoluto por unidad de paso, se puede proceder de la siguiente forma:

- si $\varepsilon_1 < \frac{|E(\overline{x})|}{h} < \varepsilon_2$, $\curvearrowright$ aceptar $v = y_{h/2}(\overline{x})$ como nueva ordenada y mantener el paso h
- si $\frac{|E(\overline{x})|}{h} > \varepsilon_2$, $\curvearrowright$ $h/2 \longrightarrow h$ reducción del paso y mantengo $v = y_{h/2}(\overline{x})$ (ya calculada) como nueva ordenada
- si $\frac{|E(\overline{x})|}{h} < \varepsilon_1$, $\curvearrowright$ $2h \longrightarrow h$ (ampliación del paso a partir de la nueva ordenada y mantener $u = y_h(\overline{x})$)

Esta estrategia es necesaria cuando en algún subintervalo de $[a, b]$ la solución $y(x)$ presenta variaciones grandes, lo cual no se sabe a priori por no conocerse $y(x)$, y sólo se detecta al analizar el error paso a paso.

Al restringir el cambio de paso a duplicaciones y reducciones a la mitad, resulta conveniente escoger la cota inferior ε_1 del error absoluto por unidad de paso como

$$\varepsilon_1 = \frac{\varepsilon_2}{2^{r+1}},$$

ya que para un método de orden r, reducir el paso a la mitad afecta el error local aproximadamente en un factor $1/2^{r+1}$. Por ejemplo, para el método de Runge-Kutta de orden 4 se tendría que $\varepsilon_1 = \varepsilon_2/2^5 = \varepsilon_2/32$.

3.2.6. Algoritmo de Runge-Kutta con cambio de paso

Sustituir el paso 5 por:

Paso 5: mientras $x < b$, repetir:
5.1) poner $x \curvearrowright \xi$, $y \curvearrowright \eta$, calcular $hf(\xi, \eta)$ y guardarlo en k_1
para $m = 2 : r$
calcular $x + c_m h \curvearrowright \xi$
calcular $\sum_{j=1}^{r-1} a_{rj} k_j \curvearrowright z$,
$y + z \curvearrowright \eta$
calcular $hf(\xi, \eta)$ y guardarlo en k_m
fin
5.2) resultados del nuevo paso:
calcular $x + h$ y guardarlo en $\overline{x}$ y x_i
calcular $\sum_{j=1}^{r} b_{rj} k_j \curvearrowright z$,
$y + z$, y guardarlo en y y en y_i
5.3) poner $y \curvearrowright u$,
hacer dos pasos con $h/2$ a partir de x para calcular v a partir de y,
o calcular por RKF45, $u = y_i^{(5)}$ y $v = y_i^{(6)}$
5.4) estimar el error absoluto:$E(\overline{x}) = \frac{2^r(v-u)}{2^r-1}$ o $E(\overline{x}) = v - u$
5.5) analizar el cambio de paso:
si h es adecuado, imprimir $\overline{x}, v$, poner $\overline{x}$ en x y x_i,y poner v en y y y_i,
si no, cambiar h
fin

La incorporación del cambio de paso al algoritmo de Runge-Kutta constituye una forma empírica de hallar el valor óptimo de h que minimice el error global (ver gráfica), supuesto que hay estabilidad numérica.

Propiedades de los métodos de Runge-Kutta

El estudio realizado de los métodos de Runge-Kutta explícitos, representativos de los de paso simple, permite resumir sus propiedades:

1- Se autoinician, es decir, basta conocer la condición inicial para poder aplicarlos

2- Para $r \leq 4$, se realizan en cada paso r evaluaciones de la parte derecha $f = y'$. Para $r \geq 5$ la cantidad de evaluaciones de función es siempre mayor que r, no obstante, en cualquier caso el costo computacional es menor que el del algoritmo de Taylor, que requiere obtener y evaluar las derivadas superiores

3- No proveen directamente forma de estimar el error, aunque puede hacerse con trabajo adicional (empleando el método de doble cómputo o la combinación de métodos de distintos órdenes, p.e. RK45)

4- El cambio de paso es fácil de realizar

5- Son generalmente estables, basta tomar h suficientemente pequeño

La propiedad 2 constituye una ventaja con respecto al algoritmo de Taylor, sin embargo, luego veremos que los métodos de paso múltiple son mejores en este sentido. La propiedad 3 es una franca

deficiencia de los métodos de Runge-Kutta.

3.3. Fórmulas de Runge-Kutta para sistemas

El problema de Cauchy para un sistema de ecuaciones diferenciales de primer orden tiene la forma:

$$\begin{cases} y_1' = f_1(x, y_1, y_2, \ldots, y_n), & y_1(a) \\ y_2' = f_2(x, y_1, y_2, \ldots, y_n), & y_2(a) \\ \qquad\qquad \cdots & \\ y_n' = f_n(x, y_1, y_2, \ldots, y_n), & y_n(a) \end{cases}, \tag{3.39}$$

$x \in [a, b]$, o en forma vectorial,

$$Y\,' = f(x, Y),\ Y(a), \tag{3.40}$$

$$Y\,' = \begin{bmatrix} y_1' \\ \vdots \\ y_n' \end{bmatrix}, \tag{3.41}$$

$$f(x, Y) = \begin{bmatrix} f_1(x, y_1, y_2, \ldots, y_n) \\ \vdots \\ f_n(x, y_1, y_2, \ldots, y_n) \end{bmatrix},\ Y(a) = \begin{bmatrix} y_1(a) \\ \vdots \\ y_n(a) \end{bmatrix}. \tag{3.42}$$

Los métodos de Runge-Kutta pueden extenderse para la resolución del problema (3.39) (justificación en Berezin-Zhidkov, vol.2), planteando la coincidencia de r términos del desarrollo en serie de Taylor de las funciones desconocidas en el punto x_i :

$$\begin{aligned} y_1(x_i + h) &= y_1(x_i) + h y_1'(x_i) + \cdots + \frac{h^r}{r!} y_1^{(r)} \\ y_2(x_i + h) &= y_2(x_i) + h y_2'(x_i) + \cdots + \frac{h^r}{r!} y_2^{(r)} \\ &\vdots \\ y_n(x_i + h) &= y_n(x_i) + h y_n'(x_i) + \cdots + \frac{h^r}{r!} y_n^{(r)} \end{aligned}$$

con expresiones de la forma:

$$\begin{aligned} y_1(x_i + h) &= y_1(x_i) + \Delta y_1(x_i) \\ y_2(x_i + h) &= y_2(x_i) + \Delta y_2(x_i) \\ &\vdots \\ y_n(x_i + h) &= y_n(x_i) + \Delta y_n(x_i) \end{aligned}$$

donde los incrementos $\Delta y_j(x_i), 1 \le j \le n$, son combinaciones lineales de las primeras derivadas y_j' evaluadas en r puntos de abscisas comprendidas en el intervalo $[x_i, x_i + h]$. Nótese que ahora la parte derecha de cada ecuación diferencial depende de más variables, a saber $y_j' = f_j(x, y_1, y_2, \ldots, y_n)$, por lo que para encontrar los desarrollos en serie de Taylor la dificultad en el cálculo de las derivadas

superiores de $y_j(x)$ se acentúa . A modo de ejemplo veamos la adaptación de una fórmula de Runge-Kutta de 4to. orden (**??**) para siguiente el problema de Cauchy:

$$\begin{aligned} y' &= f(x,y,z), \quad y(a) \\ z' &= g(x,y,z), \ z(a). \end{aligned}$$

Como el sistema consta de dos ecuaciones diferenciales estamos buscando dos funciones desconocidas, $y(x)$ y $z(x)$, y habrá que plantear dos fórmulas del tipo (**??**), una para el cálculo de cada función desconocida:

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ z_{i+1} &= z_i + \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4) \end{aligned} \tag{3.43}$$

donde

$$k_1 = h\ f(x_i, y_i, z_i), \tag{3.44}$$
$$l_1 = h\ g(x_i, y_i, z_i), \tag{3.45}$$
$$k_2 = h\ f(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}, z_i + \frac{l_1}{2}), \tag{3.46}$$
$$l_2 = h\ g(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}, z_i + \frac{l_1}{2}), \tag{3.47}$$
$$k_3 = h\ f(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}, z_i + \frac{l_2}{2}), \tag{3.48}$$
$$l_3 = h\ g(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}, z_i + \frac{l_2}{2}), \tag{3.49}$$
$$k_4 = h\ f(x_i + h, y_i + k_3, z_i + l_3), \tag{3.50}$$
$$l_4 = h\ g(x_i + h, y_i + k_3, z_i + l_3). \tag{3.51}$$

Como se aprecia hubo que definir funciones $l_1, \ldots, l_4$ semejantes a las funciones $k_1, \ldots, k_4$, sustituyendo f por g. Además, las funciones k_m y l_m tienen ahora un argumento más, el correspondiente a la función desconocida adicional $z(x)$, que se construye en forma análoga al argumento de $y(x)$ para cada m.

Para la aplicación de la fórmula (3.44)-(3.51), se evaluarán las expresiones k_m, l_m en el orden : $k_1, l_1, \ldots, k_4, l_4$, de acuerdo con el esquema de cálculo siguiente para el paso $i+1$:

i	x_i	y_i	z_i	$f(\xi,\eta,\tau)$	$g(\xi,\eta,\tau)$	hf	hg
-	$\xi_1 = x_i$	$\eta_1 = y_i$	$\tau_1 = z_i$	$f(\xi_1,\eta_1,\tau_1)$	$g(\xi_1,\eta_1,\tau_1)$	k_1	l_1
-	$\xi_2 = x_i + \frac{h}{2}$	$\eta_2 = y_i + \frac{k_1}{2}$	$\tau_2 = z_i + \frac{l_1}{2}$	$f(\xi_2,\eta_2,\tau_2)$	$g(\xi_2,\eta_2,\tau_2)$	k_2	l_2
-	$\xi_3 = x_i + \frac{h}{2}$	$\eta_3 = y_i + \frac{k_2}{2}$	$\tau_3 = z_i + \frac{l_2}{2}$	$f(\xi_3,\eta_3,\tau_3)$	$g(\xi_3,\eta_3,\tau_3)$	k_3	l_3
-	$\xi_4 = x_i + h$	$\eta_1 = y_i + k_3$	$\tau_4 = z_i + l_3$	$f(\xi_4,\eta_4,\tau_4)$	$g(\xi_4,\eta_4,\tau_4)$	k_4	l_4
$i+1$	$x_i + h$	$y_i + \Delta y_i$	$z_i + \Delta z_i$	—	—	Δy_i	Δz_i

3.3.1. Algoritmo de Runge-Kutta para sistemas de primer orden

Dado el problema de Cauchy

$$\begin{cases} y_1' = f_1(x, y_1, .., y_n) \\ y_2' = f_2(x, y_1, .., y_n) \\ \cdots \\ y_n' = f_n(x, y_1, .., y_n), \end{cases},$$

con condiciones iniciales $y_1(a),\ y_2(a), ...y_n(a)$ y $x \in [a, T]$. Para un método dado de orden r se tienen parámetros que son fijos c, A y b. Además se necesitan subrutinas auxiliares para la evaluación de las partes derechas $f_1, f_2, \ldots, f_n$ en el sistema de ecuaciones diferenciales. El paso de integración será $h = (b - a)/m$, donde m es la cantidad de pasos. Formalicemos el algoritmo.

Algoritmo 39

Algoritmo de Runge-Kutta para sistemas

Paso 1: leer A, b, c, a, T, n *y* h *o* m
Paso 2: guardar a *en* x *y en* x_o
Paso 3: leer las condiciones iniciales $Z = (y_j(a), 1 \leq j \leq n)$,
y guardarlas en Z_j *y* Y_{oj}
Paso 4: imprimir x, Z
Paso 5: para $i = 1 : m$, *repetir:*
5.1) poner x *en* ξ
para $j = 1$ *hasta* n,
poner Z_j *en* η_j,
fin
para $j = 1$ *hasta* n,
calcular $hf_j(\xi, \eta_1, \eta_2, \ldots, \eta_n) \curvearrowright k_{1j}$,
fin
para $p = 2$ *hasta* r,
calcular $x + c_p h \curvearrowright \xi$
para $j = 1$ *hasta* n,
calcular $\sum_{l=1}^{p-1} a_{pl} k_{l\ j} \curvearrowright z$, $Z_j + z \curvearrowright \eta_j$,
fin
para $j = 1$ *hasta* n,
calcular $hf_j(\xi, \eta_1, \eta_2, \ldots, \eta_n)$, *guardarlo en* k_{pj},
fin
fin
5.2) resultados del nuevo paso:
calcular $x + h$ *y guardarlo en* x *y en* x_i
para $j = 1$ *hasta* n,
calcular $\sum_{l=1}^{r} b_{rl} k_{lj} \curvearrowright z$, $Z_j + z$, *y guardarlo en* Z_j *y en* $Y_{i,j}$
fin
5.3) imprimir x, Z
fin

3.4. Resolución del problema de Cauchy de orden superior

Dado un problema de Cauchy para una ecuación diferencial de orden n explícita:

$$\begin{aligned} y^{(n)} &= f(x, y, y', \ldots, y^{(n-1)}),\ x \in [a, b] \\ & y(a),\ y'(a), \ldots, y^{(n-1)}(a). \end{aligned} \tag{3.52}$$

es necesario transformarlo antes a un sistema de primer orden para resolverlo numéricamente utilizando las fórmulas de Runge-Kutta. Se realiza un cambio de variables:

$$\begin{aligned} y &= u_1, \\ y' &= u_1' = u_2, \\ y'' &= u_2' = u_3, \\ y^{(n-1)} &= u_{n-1}' = u_n, \\ y^{(n)} &= u_n'. \end{aligned}$$

con lo cual se obtiene el sistema

$$\begin{aligned} u_1' &= u_2, & u_1(a) \\ u_2' &= u_{3,} & u_2(a) \\ u_3' &= u_{4,} & u_3(a) \\ \vdots & & \vdots \\ u_{n-1}' &= u_n, & u_{n-1}(a) \\ u_n' &= f(x, u_1, u_2, \ldots, u_n), & u_n(a) \end{aligned}$$

para el cual se adapta la fórmula de Runge-Kutta escogida.

Ejemplo: Dado el problema de Cauchy

$$y''' - y'' \operatorname{sen}(x) - y\, y' = e^x, \quad y(1) = 0,\ y'(1) = 2,\ y''(1) = -1,$$

despejando la derivada de mayor orden,

$$y''' = y'' sen(x) + yy' + e^x$$

y haciendo el cambio de variables:

$$\begin{aligned} y' &= u, \\ y'' &= u' = v, \\ y''' &= v' \end{aligned}$$

se obtiene:

$$\begin{aligned} y' &= u \equiv f_1,\ y(1) = 0 \\ u' &= v \equiv f_2,\ u(1) = 2 \\ v' &= v\ sen(x) + y\ u + e^x \equiv f_3,\ v(1) = -1, \end{aligned}$$

Supongamos que se escoge una fórmula de Runge-Kutta de orden 2

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{2}(k_1 + k_2), \\ k_1 &= hf(x_i, y_i),\ k_2 = hf(x_i + h, y_i + k_1). \end{aligned}$$

La adaptación consiste en considerar tres fórmulas semejantes a ésta para el cálculo de las nuevas ordenadas en cada paso:

$$y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2)$$
$$u_{i+1} = u_i + \frac{1}{2}(l_1 + l_2)$$
$$v_{i+1} = v_i + \frac{1}{2}(m_1 + m_2),$$

con funciones k, l, m que dependen de las partes derechas respectivas:

$$\begin{aligned} k_1 &= hf_1(x_i, y_i, u_i, v_i) = hu_i \\ l_1 &= hf_2(x_i, y_i, u_i, v_i) = hv_i \\ m_1 &= hf_3(x_i, y_i, u_i, v_i) = h[v_i\ sen(x_i) + y_i\ u_i + e^{x_i}] \end{aligned}$$

$$\begin{aligned} k_2 &= hf_1(x_i + h, y_i + k_1, u_i + l_1, v_i + m_1) \\ &= h(u_i + l_1) \\ l_2 &= hf_2(x_i + h, y_i + k_1, u_i + l_1, v_i + m_1) \\ &= h(v_i + m_1) \\ m_2 &= hf_3(x_i + h, y_i + k_1, u_i + l_1, v_i + m_1) \\ &= h[(v_i + m_1)\ sen(x_i + h) + (y_i + k_1)(u_i + l_1) + e^{x_i+h}]. \end{aligned}$$

El cálculo se hará en el siguiente orden : $k_1, l_1, m_1, k_2, l_2, m_2$, y después $y_{i+1}, u_{i+1}, v_{i+1}$ en cada paso.

Como se ha visto los métodos de Runge-Kutta realizan en cada paso varias evaluaciones de función, que hasta los métodos de orden 4 coinciden con el orden. En la sección que aparece a continuación se estudian los llamados métodos de paso múltiple que a pesar de haber sido considerados mucho antes que los métodos de Runge-Kutta, representan una simplificación de éstos al mantener el mismo orden y realizar menos evaluaciones de función en cada paso, una vez que arrancan.

3.5. Los métodos de paso múltiple

Los métodos que calculan la solución usando propiamente el desarrollo en serie de Taylor así como los métodos de Runge-Kutta requieren información en un único punto x_i, a partir del cual se obtiene y_{i+1} en el punto siguiente $x_i + h$. De ahí que se diga que estos métodos son adecuados para iniciar la resolución del problema de Cauchy.

Los métodos de paso múltiple requieren información sobre la solución en $n+1$ puntos anteriores $x_i, x_{i-1}, \ldots, x_{i-n}$ ($n \geq 1$) con vista a obtener la nueva ordenada y_{i+1}.

El objetivo de estos métodos es obtener en cada paso el valor aproximado y_{i+1} con tanta precisión como por Taylor o Runge-Kutta, pero evaluando la función $f = y'$ *menos veces*, con lo cual se alivia considerablemente el trabajo de cálculo, y se reduce por tanto el tiempo de cálculo en computadora. Por esta razón se les llama métodos para *continuar la resolución* del problema de Cauchy.

Los métodos de paso múltiple se obtienen de la integración numérica de la ecuación diferencial.

Se plantea integrar la ecuación diferencial $y' = f(x, y)$ en el intervalo $[x_{i-p}, x_{i+1}]$

$$\int_{x_{i-p}}^{x_{i+1}} y' dx = \int_{x_{i-p}}^{x_{i+1}} f(x, y(x)) dx$$

o sea,

$$y_{i+1} - y_{i-p} = \int_{x_{i-p}}^{x_{i+1}} f(x, y(x)) dx \tag{3.53}$$

Para calcular la integral que aparece en el término de la derecha es necesario disponer de aproximaciones de las funciones y y $y' = f$ en $n + 1$ puntos equidistantes para construir un polinomio de interpolación de grado n. Si se tienen nodos de interpolación $x_i, x_{i-1}, \ldots, x_{i-n}$ y los respectivos valores $f_i, f_{i-1}, \ldots, f_{i-n}$, entonces al aproximar $f(x, y)$ por un polinomio de interpolación $p_n(x)$ se obtendrá una fórmula explícita para el cálculo de y_{i+1}. Sin embargo si se consideran como nodos de interpolación $x_{i+1}, x_i, \ldots, x_{i-n+1,}$ con los respectivos valores $f_{i+1}, f_i, \ldots, f_{i-n+1}$, entonces se obtendrá una fórmula implícita. Comencemos por el caso explícito.

3.5.1. Fórmulas explícitas de Adams-Bashforth

Dados $f_i, f_{i-1}, \ldots, f_{i-n}$ se construye $p_n(x)$ y se tiene

$$y_{i+1} - y_{i-p} = \int_{x_{i-p}}^{x_{i+1}} p_n(x) dx \tag{3.54}$$

Como se observa en la expresión (3.54) hay que fijar dos parámetros, uno es p, que nos dice cuál es la solución anterior a partir de la cual se calcula la solución en el nodo x_{i+1}. El objetivo de integrar en un intervalo $[x_{i-p}, x_{i+1}]$ que contenga nodos anteriores a x_i, es aumentar la precisión del valor y_{i+1}, puesto que se calcula a partir de un y_{i-p} anterior a y_i, y por tanto, menos afectado de error que este último. Un segundo parámetro a fijar es n, según el orden del polinomio de interpolación que se considere, así será el orden de precisión de la fórmula resultante. Además recordemos que el polinomio de interpolación de grado n que pasa por $n + 1$ nodos es único, pero se puede expresar en diferentes bases; si se selecciona la base de Newton en diferencias finitas se tiene

$$p_n(x) = \sum_{k=0}^{n} \frac{1}{k! h^k} \Delta^k f_{i-k} \Pi_{j=0}^{k-1} (x - x_{i-j}) \tag{3.55}$$

e introduciendo el cambio de variable $s = \frac{x - x_i}{h}$ se tiene

$$x - x_{i-j} = (x_i + sh) - (x_i - jh) = h(s + j) \ , \tag{3.56}$$

y sustituyendo (3.56) en (3.55),

$$p_n(x) = p_n(x_i + sh) = \sum_{k=0}^{n} \frac{1}{k! h^k} \Delta^k f_{i-k} \Pi_{j=0}^{k-1} h(s + j). \tag{3.57}$$

Pero la productoria es equivalente a

$$\Pi_{j=0}^{k-1} h(s + j) = h^k (s(s + 1) \cdots (s + k - 1)),$$

y si recordamos la definición de función binomial generalizada (ref??) para $y \in \mathbb{R}$, $k \in \mathbb{N}$

$$b(y) = \left(\begin{array}{c} y \\ k \end{array} \right) = \left\{ \begin{array}{c} 1, \ \text{ si } \ k = 0 \\ \Pi_{j=o}^{k-1} \frac{y-j}{j+1} = (\frac{y}{1})(\frac{y-1}{2}) \cdots (\frac{y-k+1}{k}), \ \text{ si } k > 0 \end{array} \right. ,$$

Entonces para denotar la productoria en (3.57) de forma compacta, se hace $y - j = -(-s - j)$ y se tiene

$$\Pi_{j=0}^{k-1} \frac{(-)(-s-j)}{j+1} = (-1)^k \Pi_{j=0}^{k-1} \frac{(-s-j)}{j+1} = (-1)^k \left(\begin{array}{c} -s \\ k \end{array} \right)$$

se obtiene finalmente

$$p_n(s) = \sum_{k=0}^{n} (-1)^k \cdot \left(\begin{array}{c} -s \\ k \end{array} \right) \Delta^k f_{i-k} \ , \tag{3.58}$$

Sustituyendo (3.58) en (3.54)

$$y_{i+1} - y_{i-p} = \int_{x_{i-p}}^{x_{i+1}} \left\{ \sum_{k=0}^{n} (-1)^k \cdot \left(\begin{array}{c} -s \\ k \end{array} \right) \Delta^k f_{i-k} \right\} dx$$

Ahora es necesario efectuar el cambio de variable planteado $x = x_i + sh$, por lo que $dx = h \ ds$, y entonces los nuevos límites de integración serán: para $x = x_{i-p} \curvearrowright \ s = -p$, y para $x = x_{i+1} \curvearrowright s = 1$. Finalmente se obtiene,

$$y_{i+1} - y_{i-p} = \int_{-p}^{1} \left\{ \sum_{k=o}^{n} (-1)^k \left(\begin{array}{c} -s \\ k \end{array} \right) \Delta^k f_{i-k} \right\} h \ ds \ ,$$

de donde

$$y_{i+1} = y_{i-p} + h \int_{-p}^{1} \left\{ \sum_{k=o}^{n} (-1)^k \left(\begin{array}{c} -s \\ k \end{array} \right) \Delta^k f_{i-k} \right\} ds,$$

que puede escribirse en la forma

$$y_{i+1} = y_{i-p} + h \sum_{k=o}^{n} \alpha_k \Delta^k f_{i-k}, \tag{3.59}$$

donde

$$\alpha_k = (-1)^k \int_{-p}^{1} \left(\begin{array}{c} -s \\ k \end{array} \right) ds \ . \tag{3.60}$$

La fórmula (3.59) se conoce como fórmula de Adams-Bashforth. Nótese que los coeficientes α_k son independientes de f, por lo que pueden calcularse a partir de la función binomial $b(s)$. Los primeros son:

$$\alpha_o = 1, \ \alpha_1 = 1/2, \ \alpha_2 = 5/12, \ \alpha_3 = 3/8, \ \alpha_4 = 251/720, \ldots \tag{3.61}$$

La fórmula (3.59) es de tipo explícito con respecto a y_{i+1}, pues este valor se calcula en términos de un incremento que contiene a f_i hasta f_{i-n},

$$y_{i+1} = y_{i-p} + incr(h; f_i, \ldots, f_{i-n})$$

vale decir, en términos de y_i hasta y_{i-n}.

En la práctica, es más ventajoso desde el punto de vista computacional, una vez fijado n trabajar con ordenadas y no con diferencias finitas cuando el paso h es muy pequeño, para evitar pérdida de significación. Por ejemplo, la fórmula correspondiente a $n = 3, p = 0$ quedaría:

$$\begin{aligned} y_{i+1} &= y_i + h\textstyle\sum_{k=o}^{3}\alpha_k\Delta^k f_{i-k} \\ &= y_i + h\ [\alpha_o f_i + \alpha_1\Delta f_{i-1} + \alpha_2\Delta f_{i-2} + \alpha_3\Delta f_{i-3} \\ &= y_i + h\ [1\cdot f_i + \frac{1}{2}(f_i - f_{i-1}) + \\ &\cdot\cdot\frac{5}{12}(f_i - 2f_{i-1} + f_{i-2}) + \frac{3}{8}(f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3})] \qquad (3.62) \\ &= y_i + \frac{h}{24}[55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}] \qquad (3.63) \end{aligned}$$

Hay que destacar que en el paso $i+1$ para el cálculo de la nueva ordenada y_{i+1} sólo habría que hacer *una* evaluación de función ($f_i = f(x_i, y_i)$) y no cuatro, pues f_{i-1}, f_{i-2} y f_{i-3} ya fueron calculadas en pasos anteriores.

El *error de método o local* de las fórmulas de Adams-Bashforth se obtiene integrando el error del polinomio de interpolación. Por ejemplo, para la fórmula (3.63) el error será

$$\begin{aligned} E_3 &= h\ \alpha_4\Delta^4 f_{i-4} = h\frac{251}{720}h^4 f^{\ IV}(\xi) \\ &= \frac{251}{720}h^5 f^{\ IV}(\xi),\ x_i < \xi < x_i + h \end{aligned}$$

luego, $E_3 = O(h^5)$, y la fórmula tiene precisión de orden 4. En general, para una fórmula explícita de orden n,

$$\begin{aligned} E_n &= h\ \alpha_{n+1}\Delta^{n+1} f_{i-k} \qquad (3.64) \\ &= h^{n+2} f^{(n+1)}(\xi) = O(h^{n+2}), \qquad (3.65) \end{aligned}$$

lo que equivale a decir que la precisión es de orden $n+1$. Nótese que el error local podría evaluarse directamente teniendo en cuenta que $\Delta^{n+1} f_{i-k}$ se podría calcular usando la fórmula del binomio de Newton, por ejemplo, $\Delta^4 f_{i-4} = f_i - 4f_{i-1} + 6f_{i-2} - 4f_{i-3} + f_{i-4}$.

Estas fórmulas de paso múltiple de tipo explícito son a veces numéricamente inestables.

Ejemplos

1. $n = 0, p = 0$ se obtiene la fórmula de Euler $y_{i+1} = y_i + h\ \alpha_o f_i = y_i + h\ f_i$.

2. $p = 0$, $n = 1$

$$\begin{aligned} y_{i+1} &= y_i + h[\alpha_o f_i + \alpha_1\Delta f_{i-1}] \\ &= y_i + h[1\cdot f_i + \frac{1}{2}\cdot\Delta f_{i-1}] \\ &= y_i + h[f_i + \frac{1}{2}(f_i - f_{i-1})] \\ &= y_i + \frac{h}{2}[3f_i - f_{i-1}] \end{aligned}$$

con error

$$\begin{aligned} E_1 &= h\alpha_2\Delta^2 f_{i-2}\ ,\ \ x_i < \xi < x_i + h \\ &= h\frac{5}{12}h^2 f\,''(\xi) \\ &= \frac{5}{12}h^3(\xi) \\ &= O(h^3) \end{aligned}$$

3. $p = 1$, $n = 1$

$$y_{i+1} = y_{i-1} + h[\alpha_o f_i + \alpha_1 \Delta f_{i-1}]$$

$$\alpha_o = \int_{-1}^{1} \begin{pmatrix} -s \\ 0 \end{pmatrix} ds = 2$$

$$\alpha_1 = (-1)\int_{-1}^{1} \begin{pmatrix} -s \\ 1 \end{pmatrix} ds = 0$$

luego

$$\begin{aligned} y_{i+1} &= y_{i-1} + h[2 \cdot f_i + 0 \cdot \Delta f_{i-1}] \\ &= y_{i-1} + 2hf_i \end{aligned} \tag{3.66}$$

y la fórmula que se obtiene es la misma que si fuera $n = 0$. El error local es

$$\begin{aligned} E_1 &= h\ \alpha_2\Delta^2 f_{i-2}\ ,\ \ x_i < \xi < x_i + h \\ &= h\frac{1}{3}h^2 f\,''(\xi) = \frac{1}{3}h^3 f\,''(\xi) \\ &= O(h^3) \end{aligned}$$

donde

$$\alpha_2 = (-1)^2\int_{-1}^{1} \begin{pmatrix} -s \\ 2 \end{pmatrix} ds = \frac{1}{3}.$$

4. $p = 1$, $n = 2$:

$$\begin{aligned} y_{i+1} &= y_{i-1} + h[\alpha_o\ f_i + \alpha_1\Delta f_{i-1} + \alpha_2\Delta f_{i-2}], \\ &= y_{i-1} + h[2 \cdot f_i + 0 \cdot \Delta f_{i-1} + \frac{1}{3} \cdot \Delta^2 f_{i-2}] \\ &= y_{i-1} + h[2f_i + \frac{1}{3}(f_i - 2f_{i-1} + f_{i-2})] \\ &= y_{i-1} + \frac{h}{3}[7f_i - 2f_{i-1} + f_{i-2}] \end{aligned}$$

con error

$$\begin{aligned} E_2 &= h\ \alpha_3\Delta^3 f_{i-3}\ ,\ \ x_i < \xi < x_i + h \\ &= h\frac{1}{3}h^3 f\,'''(\xi) \\ &= \frac{1}{3}h^4 f\,'''(\xi) \\ &= O(h^4). \end{aligned}$$

Aquí, el coeficiente α_3 vale

$$\alpha_3 = (-1)^3 \int_{-1}^{1} \left(\begin{array}{c} -s \\ 3 \end{array} \right) ds = \frac{1}{3}.$$

3.5.2. Fórmulas implícitas de Adams-Moulton

Si el polinomio de interpolación de grado n lo construimos ahora con nodos $x_{i+1}, x_i, \ldots, x_{i-n+1}$,

$$p_n(x) = p_n(x_{i+1} - h + sh) = p_n(x_{i+1} + (s-1)h),$$

sustituyendo s por $s-1$ se obtiene,

$$p_n(x_{i+1} + (s-1)h) = \sum_{k=o}^{n} (-1)^k \left(\begin{array}{c} -s+1 \\ k \end{array} \right) \Delta^k f_{i+1-k} \ ,$$

lo que define fórmulas de tipo implícito conocidas como fórmulas de Adams-Moulton:

$$y_{i+1} = y_{i-p} + h \sum_{k=o}^{n} \beta_k \ \Delta^k f_{i+1-k} \ , \tag{3.67}$$

$$\beta_k = (-1)^k \int_{-p}^{1} \left(\begin{array}{c} -s+1 \\ k \end{array} \right) ds, \tag{3.68}$$

$$\tag{3.69}$$

$$\left(\begin{array}{c} -s+1 \\ k \end{array} \right) = \left\{ \begin{array}{c} 1, \text{ si } k = 0 \\ \frac{(-s+1)(-s)\cdots(-s-k+2)}{k!}, \text{ si } k > 0 \end{array} \right. .$$

Estas fórmulas son de tipo implícito porque para $k = 0$, $\Delta^o f_{i+1-o} = f_{i+1} = f(x_{i+1}, y_{i+1})$, y la nueva ordenada y_{i+1} aparece en términos de sí misma.

Los primeros valores de los coeficientes β son: $\beta_o = 1$, $\beta_1 = -1/2$, $\beta_2 = -1/12$, $\beta_3 = -1/24$, $\beta_4 = -19/720$,

El error de método de las fórmulas implícitas se obtiene, de forma análoga a como se obtuvo para las fórmulas explícitas, integrando el error del polinomio de interpolación. Para la aproximación de f por un polinomio de grado n,

$$\begin{aligned} E_n &= h\beta'_{n+1} \Delta^{n+1} f_{i+1-k} \\ &= h\beta'_{n+1} h^{n+1} f^{(n+1)}(\xi) \\ &= h^{n+2} \beta'_{n+1} f^{(n+1)}(\xi) = O(h^{n+2}), \end{aligned}$$

es decir, el error local es de orden $n+2$. Se dice entonces que la correspondiente fórmula de paso múltiple tiene precisión de orden $n+1$.

Ejemplos

1. La fórmula más sencilla de paso múltiple de tipo implícito se obtiene para $p = 0$, $n = 1$, que constituye la que se conoce como fórmula trapezoidal

$$\begin{aligned} y_{i+1} &= y_i + h[1 \cdot f_{i+1} - \frac{1}{2} \cdot \Delta f_i] \\ &= y_i + h[f_{i+1} - \frac{1}{2}(f_{i+1} - f_i)] \\ &= y_i + \frac{h}{2}[f_{i+1} + f_i] \end{aligned} \tag{3.70}$$

con error

$$E_1 = h\ \beta_2\ \Delta^2 f_{i+1-2} = h(-\frac{1}{12})h^2 f\ ''(\xi) = -\frac{1}{12}h^3 f\ ''(\xi), \quad x_i < \xi < x_i + h$$
$$E_1 = O(h^3)$$

2. $p = 0$, $n = 3$:

$$\begin{aligned} y_{i+1} &= y_i + h[\beta_o f_{i+1} + \beta_1 \Delta f_i + \beta_2 \Delta^2 f_{i-1} + \beta_3 \Delta^3 f_{i-2}] \\ &= y_i + h[1 \cdot f_{i+1} + (-\frac{1}{2}) \cdot (f_{i+1} - f_i) + (-\frac{1}{12}) \cdot (f_{i+1} - 2f_i + f_{i-1}) \\ &+ (-\frac{1}{24}) \cdot (f_{i+1} - 3f_i + 3f_{i-1} - f_{i-2})] \qquad (3.71) \\ &= y_i + \frac{h}{24}[9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}] \qquad (3.72) \end{aligned}$$

con error

$$E_3 = h\beta_4 \Delta^4 f_{i+1-4} = h(-\frac{19}{720})h^4 f^{\ IV}(\xi) = -\frac{19}{720}h^5 f^{\ IV}(\xi), \quad x_i < \xi < x_i + h$$
$$E_3 = O(h^5)$$

Nótese que esta fórmula tiene precisión de orden 4, pero con un coeficiente de módulo menor que la explícita (3.63) correspondiente a los mismos valores de p y n.

Uso de las fórmulas implícitas

Para calcular y_{i+1} mediante fórmulas implícitas, es necesario establecer un proceso iterativo de la forma

$$y_{i+1}^{(j+1)} = y_{i-p} + incr(h; f(x_{i+1}, y_{i+1}^{(j)}), f_i, \ldots, f_{i-n+1})$$

o sea,

$$y_{i+1}^{(j+1)} = y_{i-p} + h\sum\nolimits_{k=o}^{n} \beta_k\ \Delta^k f_{i+1-k}^{(j)},$$

y sustituyendo las diferencias $\Delta^k f_{i+1-k}$ en términos de $f_{i+1}, f_i, \ldots, f_{i+1-n}$ se obtiene

$$\begin{aligned} y_{i+1}^{(j+1)} &= y_{i-p} + hb\ f(x_{i+1}, y_{i+1}^{(j)}) + h\sum\nolimits_{k=1}^{n} b_k f_{i+1-k} \qquad (3.73) \\ &\equiv g(y_{i+1}^{(j)}) \end{aligned}$$

donde b es el coeficiente numérico del valor $f_{i+1}^{(j)}$ que contiene la aproximación anterior de la ordenada $y_{i+1}^{(j)}$:

$$b = \sum_{k=o}^{n} \beta_k \ ,$$

y los coeficientes $b_k, 1 \leq k \leq n$, son los de los valores $f_{i+1-k} = f(x_{i+1-k}, y_{i+1-k})$ que no varían con la iteración. La condición suficiente de convergencia del proceso iterativo (3.73) será entonces, según el teorema del punto fijo,

$$|g'(y_{i+1})| < 1$$

es decir,

$$\left| h \ b \ \frac{\partial f}{\partial y_{i+1}} \right| \leq h \, |b| \, K < 1,$$

donde

$$K = \max_{a \leq x \leq b} \left| \frac{\partial f}{\partial y}(x, y(x)) \right| \ ,$$

lo que impone la condición

$$h < \frac{1}{|b| \, K} \tag{3.74}$$

También será necesario disponer de una aproximación inicial $y_{i+1}^{(o)}$ para poder aplicar (3.73). Se podría tomar $y_{i+1}^{(o)} = y_i$, pero una buena aproximación inicial es también decisiva para garantizar la convergencia del proceso iterativo. De ahí que sea más conveniente utilizar como $y_{i+1}^{(o)}$ el valor provisto por una fórmula explícita de paso múltiple, preferiblemente del mismo orden que la implícita que se use. La necesidad de esta combinación dio lugar a los llamados esquemas de predicción-corrección.

3.6. El esquema predictor-corrector

El uso de fórmulas implícitas de paso múltiple en el cálculo de y_{i+1} al resolver numéricamente el problema de Cauchy, tiene la ventaja de proporcionar valores más precisos, en general, que si se usan fórmulas explícitas. Como se vio anteriormente, esto se debe a que el coeficiente del error en módulo es menor en general, siendo igual la potencia de h que aparece en la expresión del error de método.

Sin embargo, para utilizar las fórmulas implícitas se requiere definir un proceso iterativo que sea convergente y que en muy pocos pasos (preferiblemente uno solo) dé la precisión deseada para la nueva ordenada y_{i+1}. Esto trae como consecuencia la necesidad de :

- escoger la aproximación inicial $y_{i+1}^{(o)}$ convenientemente
- escoger h en cada paso, de modo que se cumpla la condición de convergencia vista más arriba para la fórmula ímplicita.

La primera exigencia se puede satisfacer fácilmente, escogiendo para el cáculo de $y_{i+1}^{(o)}$ una fórmula explícita del mismo orden que la implícita. La segunda exigencia es más difícil de satisfacer, pero veremos cómo, introduciendo una estrategia de cambio de paso en el llamado esquema predictor-corrector, esto puede lograrse.

Esquema predictor-corrector

El uso combinado de fórmulas explícitas e implícitas de paso múltiple, de modo que el valor y_{i+1} obtenido por una fórmula explícita se tome como predicción o valor inicial $y_{i+1}^{(o)}$ y se corrija mediante una fórmula implícita hasta obtener una aproximación $y_{i+1}^{(j+1)}$ suficientemente buena de y_{i+1}, se denomina *esquema predictor-corrector.* A continuación se verá cómo implementar un esquema predictor-corrector para el problema de Cauchy de primer orden $y' = f(x, y)$, $y(a)$, $x \in [a, b]$.

A partir del punto inicial $x_o = a$, $y_o = y(a)$ se evalúa $f_o = f(x_o, y_o)$, y utilizando una fórmula de Runge-Kutta de orden n se calcula el número mínimo k de valores y_i necesarios para poder aplicar el esquema predictor-corrector de parámetros p y n, con paso de integración h constante, según se describe a continuación.

Algoritmo 40

Esquema predictor-corrector

Paso 1: leer $a, b, y(a), N$ *(número de divisiones del intervalo [a,b]),*
calcular el paso de integración $h = \frac{b-a}{N}$
Paso 2: $a \curvearrowright x_o$ *y* $y(a) \curvearrowright y_o$*, e imprimir* x_o, y_o
Paso 3: calcular y_i $(1 \le i \le k = \max(p, n))$ *por una fórmula de Runge-Kutta de orden* $\ge n$*, imprimir* x_i, y_i
Paso 4: evaluar $f(x_i, y_i)$, $0 \le i \le k$, *y poner en* f_i
Paso 5: para $i = k$ *: c repetir*
5.1 P: $0 \to j$ *y aplicar una fórmula predictora al cálculo de* $y_{i+1}^{(j)}$
5.2 E: evaluar $f(x_i + h, y_{i+1}^{(j)}) \curvearrowright f_{i+1}$
5.3 C: aplicar una fórmula correctora de orden n al cálculo de $y_{i+1}^{(j+1)}$ *usando* f_{i+1}
5.4 mientras $\left|y_{i+1}^{(j+1)} - y_{i+1}^{(j)}\right| > \varepsilon \left|y_{i+1}^{(j+1)}\right|$ *para un* ε *prefijado, poner* $j + 1 \curvearrowright j$
repetir 5.2 y 5.3
5.5 E: evaluar $f(x_i + h, y_{i+1}^{(j+1)}) \curvearrowright f_{i+1}$
imprimir x_{i+1}, y_{i+1}
fin

En el algoritmo así descrito, el ciclo correspondiente al paso 5 equivale al proceso iterativo $P(EC)^j E$,donde se hacen $j+1$ evaluaciones de la función f en cada paso. Pero aunque el valor de h utilizado garantice la convergencia, puede violarse el objetivo fundamental de las fórmulas de paso múltiple, consistente en hacer menos evaluaciones de función que las del tipo de Runge-Kutta del mismo orden por paso. De ahí la importancia de proceder de otra manera que permita mantener dicho objetivo. Esta consistiría en sustituir el paso 5.4) por hacer una estimación del error local que pueda ser usada para decidir si el valor de h es adecuado, y cambiarlo si no lo es, antes de calcular la nueva y, con lo cual el proceso iterativo correspondiente sería $P(EC)E$, con sólo dos evaluaciones de función f por paso, y h variable.

Ejemplos

1. de 2do. orden, predictor: $y_{i+1} = y_i + h\ f(x_i, y_i)$

corrector: $y_{i+1} = y_i + \frac{h}{2}[f(x_{i+1}y_{i+1}) + f(x_i, y_i)$

La fórmula predictora es la de Euler, que en realidad no es de paso múltiple, y la correctora es la fórmula trapezoidal

2. de 4to. orden de Adams-Moulton

   predictor: $y_{i+1} = y_i + \frac{h}{24}[55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}]$, $E_3 = -\frac{19}{720}h^5 f^{\ IV}(\xi)$,

   corrector: $y_{i+1} = y_i + \frac{h}{24}[9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}]$, $E_3 = -\frac{19}{720}h^5 f^{\ IV}(\xi)$,

3. de 4to. orden de Adams-Milne

   predictor: $y_{i+1} = y_{i-3} + \frac{4h}{3}[2f_i - f_{i-1} + 2f_{i-2}]$, $E_2 = \frac{28}{90}h^5 f^{\ IV}(\xi)$

   corrector: $y_{i+1} = y_{i-1} + \frac{h}{3}[f_{i+1} + 4f_i + f_{i-1}]$, $E_2 = -\frac{1}{90}h^5 f^{\ IV}(\xi)$

Estimación del error local y cambio de paso

Además de mejorar la precisión, la fórmula correctora permite obtener una estimación del error local o de discretización, que puede ser utilizada para determinar si el paso h es el adecuado para lograr la precisión requerida.

Con vista a analizar el procedimiento de estimación del error para el par predictor-corrector correspondiente a las fórmulas de 4to. orden de Adams-Bashforth y Adams-Moulton (esquema de Adams-Moulton de 4to. orden), escribimos la estimación del error local de cada una

$$E_{AB} = \frac{251}{720}h^5 f^{IV}(\xi)$$
$$E_{AM} = -\frac{19}{720}h^5 f^{IV}(\xi).$$

Sean $y_{i+1}^{(o)}$ y $y_{i+1}^{(1)}$ las aproximaciones de y_{i+1} obtenidas por la fórmula explícita y el primer paso de la implícita respectivamente. Entonces se tiene para el valor exacto de y en el punto x_{i+1} las siguientes estimaciones de error:

$$y(x_{i+1}) - y_{i+1}^{(o)} = \frac{251}{720}h^5 f^{IV}(\xi_1)$$
$$y(x_{i+1}) - y_{i+1}^{(1)} = -\frac{19}{720}h^5 f^{IV}(\xi_2) \tag{3.75}$$

donde $\xi_1 \neq \xi_2$ en general. Si suponemos que $f^{IV}(\xi)$ no varía mucho sobre el intervalo de interés, restando las expresiones anteriores obtenemos

$$y_{i+1}^{(1)} - y_{i+1}^{(o)} \approx \frac{270}{720}\ h^5 f^{\ IV}(\xi),$$

de donde despejando,

$$h^5 f^{\ IV}(\xi) \approx \frac{720}{270}(y_{i+1}^{(1)} - y_{i+1}^{(o)}),$$

y sustituyendo esta expresión en (3.75)

$$y(x_{i+1}) - y_{i+1}^{(1)} \approx -\frac{19}{720} \cdot \frac{720}{270}(y_{i+1}^{(1)} - y_{i+1}^{(o)})$$
$$\approx -\frac{1}{14}(y_{i+1}^{(1)} - y_{i+1}^{(o)}) = D_{i+1}, \tag{3.76}$$

o sea, el error del valor corregido es aproximadamente igual a $-1/14$ de la diferencia entre el valor corregido y el predicho.

Como se mencionó anteriormente, en el esquema predictor-corrector es aconsejable usar la fórmula correctora una sola vez. Así, si para el valor de h utilizado no se logra que el error estimado D_{i+1} por unidad de paso sea suficientemente pequeño, es más conveniente reducir h que realizar más iteraciones con la fórmula correctora.

En un programa general para la resolución del problema de Cauchy, la estimación del error local se utiliza de la siguiente forma. En el algoritmo presentado para h constante, efectuemos el paso 5.3) una sola vez para obtener $y_{i+1}^{(1)}$, y sustituyamos 5.4) por el cálculo de D_{i+1} según (3.76), seguido de una estrategia de cambio de paso semejante a la descrita para los métodos de Runge-Kutta y basada en el análisis del error absoluto por unidad de paso, dadas cotas ε_1 y ε_2:

- $\varepsilon_1 < \frac{|D_{i+1}|}{h} < \varepsilon_2$, $\curvearrowright$ aceptar $y_{i+1}^{(1)}$ como nueva ordenada y_{i+1},y mantener el paso h
- si $\frac{|D_{i+1}|}{h} > \varepsilon_2$, $\curvearrowright$ $h/2 \longrightarrow h$ (reducción del paso y recalcular la nueva ordenada y_{i+1})
- si $\frac{|D_{i+1}|}{h} < \varepsilon_1$, $\curvearrowright$ $2h \longrightarrow h$ (ampliación del paso a partir de la nueva ordenada y_{i+1})

Es costumbre efectuar el cambio de paso en el esquema predictor-corrector de Adams-Moulton, cuando se detecta la necesidad de hacerlo, recomputando 3 valores a partir de y_i con la fórmula de Runge-Kutta de 4to. orden mencionada, y recomenzar luego la aplicación del esquema predictor-corrector. En este caso, el paso 5 debe convertirse de ciclo for en ciclo $while x_i < b$.

Adaptación del esquema predictor-corrector para sistemas de 1er. orden

Para el problema de Cauchy correspondiente a un sistema de ecuaciones diferenciales de 1er. orden:

$$\begin{aligned} y_1' &= f_1(x, y_1, \ldots, y_n), \;\; y_1(a) \\ y_2' &= f_2(x, y_1, \ldots, y_n), \;\; y_2(a) \\ &\vdots \qquad\qquad\qquad\quad \vdots \\ y_n' &= f_n(x, y_1, \ldots, y_n), \;\; y_n(a), \end{aligned}$$

la adaptación consiste en agregar un segundo índice k $(1 \leq k \leq n)$ que identifique la ecuación. Por ejemplo, si se usa el esquema predictor-corrector de Adams-Moulton, quedaría:

$$\text{predictor:} \;\; y_{i+1,k} = y_{i,k} + \frac{h}{24}[55f_{i,k} - 59f_{i-1,k} + 37f_{i-2,k} - 9f_{i-3,k}]$$
$$\text{corrector:} \;\; y_{i+1,k} = y_{i,k} + \frac{h}{24}[9f_{i+1,k} + 19f_{i,k} - 5f_{i-1,k} + f_{i-2,k}] \; .$$

La estimación del error local y el cambio de paso se efectúan en forma similar, con sólo considerar la variable y como vector de n componentes, y sustituir el módulo por una norma vectorial.

Comparación de los métodos de Runge-Kutta y los de paso múltiple

Al estudiar los métodos de Runge-Kutta mencionamos sus propiedades, que ahora podemos comparar con las de los de paso múltiple (PM, a saber

1. Los de RK se autoinician, mientras que los de PM, no, y requieren del auxilio de los primeros para poder iniciar su aplicación, aunque son más adecuados para continuar la resolución del problema de Cauchy

2. Los de PM realizan menos evaluaciones de la función f que los de RK de un mismo orden r, que en cada paso hacen al menos r evaluaciones

3. Los de PM permiten estimar el error fácilmente y los de RK requieren de cálculos adicionales para hacerlo

4. El cambio de paso es fácil de realizar en los de RK, mientras que en los de PM es más complicado

5. Los de RK son generalmente estables, basta tomar h suficientemente pequeño, mientras que los de PM no siempre son estables numéricamente

Como se puede apreciar, las propiedades 1, 4 y 5 constituyen ventajas de los métodos de Runge-Kutta sobre los de paso múltiple, mientras que con respecto a las propiedades 2 y 3, los de paso múltiple son mejores. De aquí que la elección dependerá del problema concreto a resolver. Por ejemplo, si el intervalo de integración es muy grande, la estabilidad será un factor de peso en contra de los de paso múltiple y si la solución es muy oscilante que requiera cambios de paso frecuentes, también están éstos en desventaja aunque estimen el error más fácilmente. Pero si se trata de un problema de Cauchy para un sistema de muchas ecuaciones diferenciales y el intervalo de integración es largo, entonces el peso de las evaluaciones de función requeridas pone en desventaja a los métodos de Runge-Kutta.

3.7. Ejercicios para el estudio independiente

1. Dado el problema de Cauchy

$$\begin{aligned} y' =& \frac{y+x}{y-x} \\ y(0) =& 1 \end{aligned}$$

cuya solución exacta es $y = x + \sqrt{1+2x^2}$

   a) Calcule y(0.5) usando el método de Euler explícito, con pasos $h = 0{,}1, h = 0{,}05, h = 0{,}025$. Muestre los valores en una tabla donde refleje a su vez el valor de la solución exacta en cada uno de los pasos. Grafique los resultados.

   b) Determine el error cometido en el primer paso para cada h.

   c) Analice qué relación existe entre el error en el cálculo del último valor para los diferentes valores de h

2. Dado el problema de Cauchy

$$y' = \frac{4x}{y}$$

con condición inicial $y(0) = 1, h = 0{,}2$

   a) Calcule a mano un paso usando el mótodo de Euler explícito, Euler implícito y el método de Runge-Kutta de orden 2 para $\Theta = 1$ (RK2).

   b) Implemente una función EDO.m que resuelva la ecuación usando los métodos de Euler explícito, Euler implícito y RK2 en el intervalo [0,2].

   c) Construya un esquema predictor corrector con los métodos explícito e implícito de Euler, calcule una iteración y compare con una iteración de los métodos explícito e implícito por separado. . Copiar los resultados en una tabla

   d) Compare los resultados obtenidos por todas las vías anteriores si la solución exacta es $y = \sqrt{4x^2 + 1}$

3. Dado el siguiente sistema de EDO's que representa la reacción entre dos sustancias químicas, donde y, z, w representan las concentraciones de dichas sustancias y las constantes k_1 y k_2 representan tasas de reacciones

$$\begin{aligned}\frac{dy}{dx} &= -k_1 y - k_2 yz \\ \frac{dz}{dx} &= k_1 y - k_2 yz \\ \frac{dw}{dx} &= k_2 yz\end{aligned}$$

para valores de las constantes $k_1 = 0{,}0222$ y $k_2 = 0{,}0723$ y valores iniciales de las variables $y_0 = 1{,}6433$; $z_0 = w_0 = 0{,}0000$; $x_0 = 0{,}0$.

   a) Calcular aplicando un método de Runge Kutta de orden 2 con cuatro cifras significativas exactas y con algún comando de MATLAB de las ODES, p.e. ode23t, ode23tb

   b) Mostrar los valores obtenidos para y, z, w , cuando x toma valores de 0,0 hasta 20 con paso 0,5

4. Modelo de la propagación de una llama de fuego (de Larry Shampine[5]). Si se enciende un fósforo, la llama crece rápidamente hasta que alcanza su dimensión crítica. Luego permanece con ese tamaño, ya que la cantidad de oxígeno consumida por la combustión en el interior de la llama balancea la cantidad disponible a través de la superficie. El modelo que representa dicho proceso viene dado por la ecuación diferencial de primer orden

$$y' = y^2 - y^3$$

con $y(0) = \delta$, $0 \leq t \leq (2/\delta)$. La magnitud escalar $y(t)$ representa el radio de la llama. Los términos y^2, y^3 provienen del área de la superficie y del volumen. El parámetro δ es el radio inicial de la llama el cual es pequeño y será el parámetro crítico. Tomando varios valores de δ, comenzando con $\delta = 0{,}01$:

[5]ver NC Matlab de Moler

a) Implemente el método de Runge-Kutta de orden 2 para $\Theta = 1/2$ (Heun) tomando como paso $h = \frac{1}{\delta}$ y orden de precisión $\epsilon = 10^{-4}$.

b) Implemente Adams-Bashforth para n=3

c) Plotee la solución

d) Compare los resultados obtenidos en los dos primeros incisos con las funciones del Matlab específicas

5. Dadas dos especies de animales, donde $u(t)$ denota la densidad de los depredadores en el tiempo t y $v(t)$ la densidad de las presas. La interacción en el tiempo entre estas especies viene dada por un modelo muy conocido de la biología-matemática llamado modelo de Lotka-Volterra. Implemente los métodos de Euler explícito e implícito para resolver el siguiente caso particular. Grafique la solución y concluya

$$u' = u(v-2) \tag{3.77}$$
$$v' = v(1-u) \tag{3.78}$$
$$u(0) = 4, v(0) = 2 \tag{3.79}$$
$$\tag{3.80}$$

Capítulo 4

Aproximación de funciones por mínimos cuadrados

La teoría de aproximación de funciones como su nombre lo indica se ocupa de: dada una expresión analítica o una muestra discreta de valores $(x_0, f_0), (x_1, f_1), ..., (x_n, f_n)$ que representan una función $f(x)$; encontrar otra función $\widehat{F}(x; \overrightarrow{c})$ que la aproxime lo mejor posible dentro de una clase de funciones Φ prefijada. Los problemas más simples son dos: el primero, que ya hemos estudiado, considera que la precisión de los datos es suficiente como para que se puedan considerar exactos y obliga a que la función aproximante pase exactamente por los valores conocidos y se conoce como aproximación por interpolación. La clase de funciones más utilizada para interpolar es la de los polinomios. En este caso, se demostró que a pesar de tener una muestra de datos exactos, cuando este conjunto es muy grande, no es de utilidad considerar un solo polinomio que interpole todos los datos; aquí es conveniente usar la interpolación por tramos. Ahora bien, si además de contar con muchos datos estos se consideran afectados de error o si se conoce la expresión analítica de la función, pero es muy irregular como se muestra en la Figura (4.1), entonces no tiene sentido obligar a que la función de aproximación pase exactamente por estos valores. Es decir, en estos casos no tiene sentido aplicar interpolación.

Sin embargo, según la tendencia del comportamiento de los datos, por lo general se puede proponer la forma que tendrá la función de aproximación. En la Figura (4.2) un polinomio lineal, en la Figura (4.3) un polinomio cuadrático; si se conoce que los datos tienen un comportamiento periódico entonces en general se propone aproximar por una función trigonométrica y así según el caso.

Hasta ahora se ha hablado de la forma en que vienen dados los datos y de la clase de funciones donde se elige la función aproximante, sin embargo otro aspecto importante a tener en cuenta es el error de aproximación $(f - \hat{F})$ y la norma en la que se mide este error. Para formalizar el problema de aproximación en un contexto bien general se define un espacio lineal normado L y un subespacio Φ, $\Phi \subset L$. Entonces, dada $f \in L$, el problema de aproximación consiste en determinar la función $\widehat{F} \in \Phi$ que más cerca esté de f según

$$\left\| f - \widehat{F} \right\| = \min_{F \epsilon \Phi} \|f - F\| . \tag{4.1}$$

Si el error de aproximación (residual) se mide según la norma Euclideana, el problema de aproximación se conoce como un problema de mínimos cuadrados. Considerar otras normas para medir

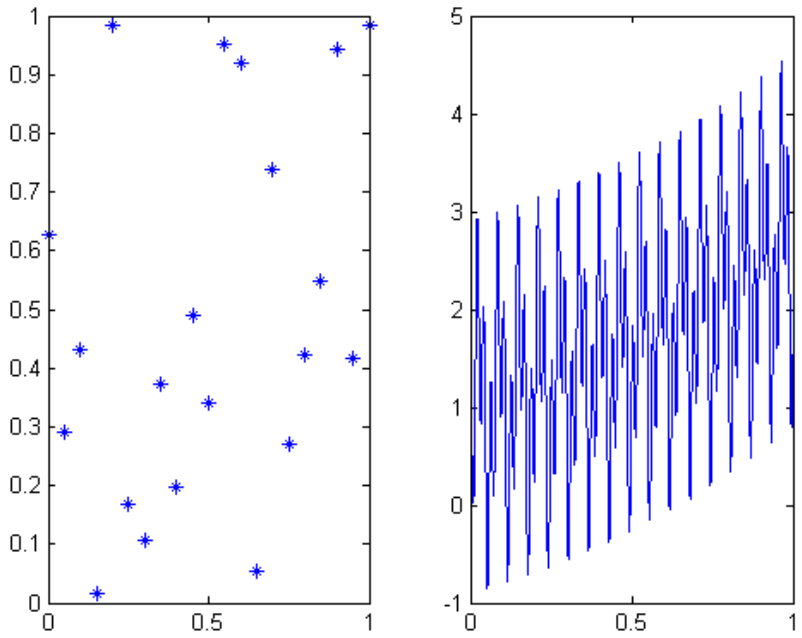


Figura 4.1:

el error conduce a otros tipos de aproximación que no trataremos aquí, consultar p.e., *Numerical Analysis. Second* Edition. Walter Gautschi. Birkhauser 2012. En las aplicaciones, el problema de aproximación mínimo cuadrática se puede encontrar relacionado con:

- Resolver $Ax = b$, $A_{m\times n}$, con $m > n$, implica $min_x \|Ax - b\|_2$.
- El ajuste de curvas.
- La modelación estadística de datos con ruido.
- La modelación geodésica.
- El problema de optimización sin restricciones.
- La estimación de parámetros.

Si se define el funcional $\varphi(F) = \|f - F\|$, $\varphi : \Phi \to \mathbb{R}$, estamos ante un problema de optimización formulado de manera abstracta y para asegurar la existencia y unicidad del valor extremo existen resultados teóricos que pueden ser consultados en el apéndice [1].

A continuación, se considerará el problema de mínimos cuadradros desde el punto de vista del ajuste de curvas.

---

[1]

- Para garantizar la existencia del producto escalar se trabaja con espacios de Hilbert.
- Los espacios Euclídeos completos de dimensión infinita con el producto escalar ordinario son espacios de Hilbert.

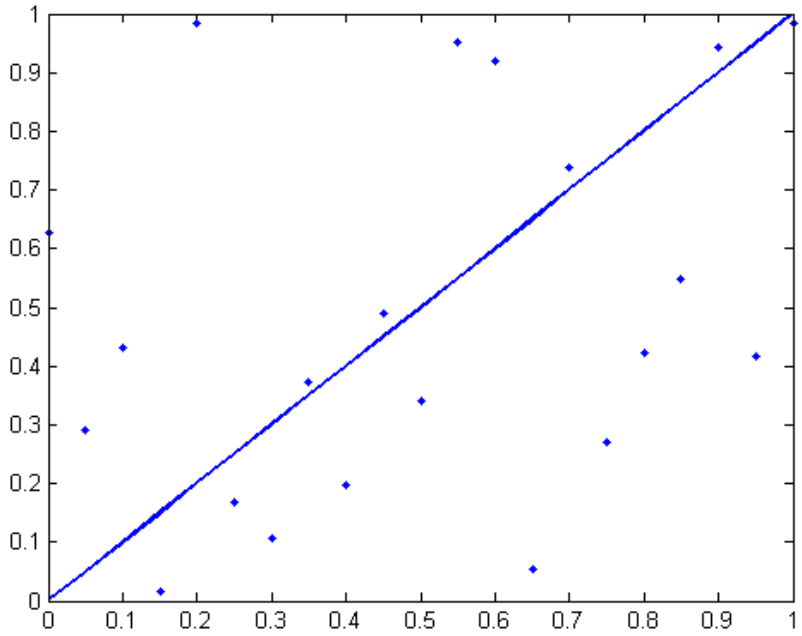


Figura 4.2:

4.1. Ajuste de curvas

Supongamos que se tiene una función $f : \mathbb{R} \to \mathbb{C}$ y se quiere encontrar $\widehat{F} \in \Phi$ que mejor aproxime a f en el conjunto de funciones

$$\Phi = \left\{F(x,c) : \mathbb{R} \to \mathbb{C}, c \in \mathbb{R}^{n+1}\right\}$$

Definición 41 *Dada una función f y una familia de funciones Φ, determinadas a priori, la función $\widehat{F} \in \Phi$, es la aproximación mínimo cuadrática de f si existen parámetros $c^* = (c_i^*)_{i=0,...,n}$, tales que*

$$r_{min} = \left\| f - \widehat{F}(x,c^*) \right\|_2 = min_{c\in\mathbb{R}^{n+1}} \|f - F(x,c)\|_2$$

En el caso de la norma discreta, se tiene

$$\left\| f - \widehat{F}(x,c^*) \right\|_2 = min_{c\in\mathbb{R}^{n+1}} \left[\sum_{i=0}^{N} (f(x_i) - F(x_i, c_0, c_1, ..., c_n))^2 \right]^{\frac{1}{2}}$$

El tratamiento del problema y las vías de solución están ahora relacionados con el hecho de que

- El funcional definido por $\varphi(F) = \|f - F\|_2^2$ es continuo, es lineal (definido sobre un espacio de Hilbert es por tanto también acotado) y es fuertemente convexo sobre un conjunto convexo B ($\rho_\varphi(x,y) := \frac{1}{2}\varphi(x) + \frac{1}{2}\varphi(y) - \varphi(\frac{x+y}{2}) \geq \gamma \|x-y\|^2$, para $x, y \in B$, $\gamma > 0$.)

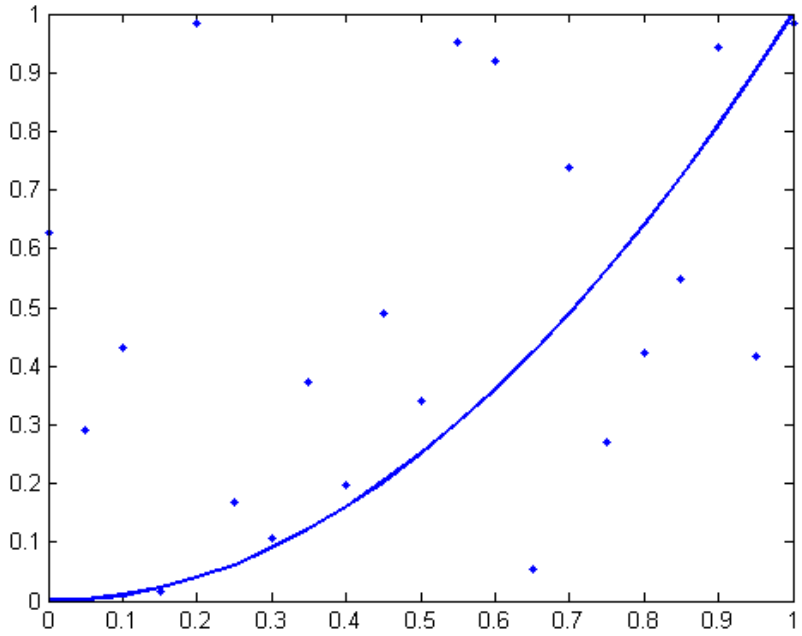


Figura 4.3:

- la función $F(x, c_0, ..., c_n)$, depende linealmente de los c_i.
- la función $F(x, c_0, ..., c_n)$, no depende linealmente de los c_i.

Veamos entonces cómo resolver el problema (4.1) para el caso particular en que la función $F(x, c), c \in \mathbb{R}^{n+1}$ depende linealmente de los parámetros c_i.

4.1.1. Ajuste de curvas lineal

Supongamos que se tiene una función $f : \mathbb{R} \to \mathbb{C}$ y se quiere encontrar $\widehat{F} \in \Phi$ que mejor aproxime a f en el conjunto de funciones

$$\Phi = \left\{F(x, c) : \mathbb{R} \to \mathbb{C}, c \in \mathbb{R}^{n+1}\right\}$$

donde $F(x, c)$ es de la forma

$$F(x, c) = \sum_{j=o}^{n} c_j \varphi_j(x),$$

siendo $\{\varphi_j(x)\}_{j=o}^{n}$ un conjunto de funciones linealmente independientes conocidas y $F(x)$ depende linealmente de los coeficientes desconocidos c_j; entonces estamos ante un problema de aproximación mínimo cuadrática lineal. El subespacio Φ es de dimensión finita y está generado por $\{\varphi_k\}_{k=0}^{n}$. El problema a resolver es

$$\left\| f - \sum_{j=o}^{n} c_j^* \varphi_j(x) \right\|_2^2 = \min_{c \in \mathbb{R}^{n+1}} \left\| f - \sum_{j=o}^{n} c_j \varphi_j(x) \right\|_2^2 \tag{4.2}$$

Se considera que la norma fue inducida por un producto escalar, $\|f\| = \sqrt{\langle f, f\rangle}$, definido según:

- producto escalar continuo $\langle f, g\rangle = \int_a^b f(x)\overline{g(x)}dx$
- producto escalar discreto $\langle f, g\rangle = \sum_{i=1}^n f(x_i)\overline{g(x_i)}$

El problema (4.2) se reduce a encontrar

$$\min_{c\in\mathbb{R}^{n+1}} \langle f - F(x,c), f - F(x,c)\rangle .$$

Note que, si denotamos $g(c) = \|f - F(x,c)\|_2$, entonces $g(c) \geq 0$, y $(g(c))^2$ será una función monótona creciente, por tanto,

$$ming(c) \Leftrightarrow min(g(c))^2$$

$$\left\|f - \widehat{F}(x,c^*)\right\|_2 = \min_{c\in\mathbb{R}^{n+1}} \|f - F(x,c)\|_2 , \tag{4.3}$$

es decir, se quiere hallar una función aproximante que minimice la norma Euclideana de la función error

$$\left\|f - \widehat{F}(x,c^*)\right\|_{2,\omega} = \left(\int_a^b \omega(x)[f(x) - \widehat{F}(x,c)]^2 dx\right)^{1/2}, \text{ en el caso continuo,}$$

$$\left\|f - \widehat{F}(x,c^*)\right\|_{2,\omega,M} = \left(\sum_{i=1}^m \omega(x_i)[f(x_i) - \widehat{F}(x_i,c)]^2\right)^{1/2}, \text{ en el caso discreto.}$$

Es importante observar que la elección de la función de peso $\omega(x)$ y los pesos $\omega(x_i)$, afecta a $\widehat{F}$. En el caso *continuo,* con una elección adecuada se puede forzar a que $\widehat{F}$ aproxime mejor a f en una parte de $[a,b]$, que en el resto del intervalo, veamos:

- $\omega(x) = 1$ en $[a,b]$, asigna igual peso a los valores de la función error para todo $x \in [a,b]$.
- $\omega(x) = 1/\sqrt{1-x^2}$ en $(-1,1)$, asigna mayor peso al error cerca de $x = -1$ y $x = 1$.
- $\omega(x) = e^{-x}$ en $[0,\infty)$, asigna peso máximo al error en $x = 0$, decreciente cuando $x \longrightarrow \infty$.
- $\omega(x) = e^{-x^2}$ en $(-\infty,+\infty)$, asigna peso máximo al error en $x = 0$, decreciente cuando $x \longrightarrow \pm\infty$.

En el caso discreto, un valor grande $\omega_i = \omega(x_i)$ significa que al valor del error $f_i - F_i$ se le confiere mucha importancia porque f_i fue medido con gran precisión y un valor ω_i pequeño es indicador de poca confiabilidad en el valor f_i (en la terminología estadística, se dice que (x_i, f_i) es un punto ruidoso o *outlier*). Nosotros comenzaremos considerando el caso discreto con $\omega(x) = 1$. El siguiente teorema es la base para la determinación de la mejor aproximación mínimo cuadrática lineal $\widehat{F}$, tanto en el caso continuo, como en el discreto.

Teorema 42 *Sean las funciones $\varphi_o, \varphi_1, ..., \varphi_n$ linealmente independientes y que generan al subespacio Φ. Entonces existe una función única $\widehat{F}$ de la forma $\widehat{F} = \sum_{j=o}^{n} c_j^* \varphi_j$, tal que*

$$\left\| f - \widehat{F} \right\|_2^2 \leq \|f - F\|_2^2, \ \forall F = \sum_{j=o}^{n} c_j \varphi_j,$$

$\widehat{F}$ es también solución del sistema de ecuaciones lineales que se obtiene resolviendo las ecuaciones normales:

$$\langle f - \widehat{F}, \varphi_k \rangle = 0, \quad 0 \leq k \leq n$$

y viceversa.

Demostración 43 *Teniendo en cuenta las propiedades del producto escalar y la forma de $\widehat{F}$,*

$$\begin{aligned} &< f - \widehat{F}, \varphi_k >= 0 \ , \ 0 \leq k \leq n \\ &\Leftrightarrow < \widehat{F}, \varphi_k >=< f, \varphi_k > \\ &< \sum_{j=o}^{n} c_j^* \varphi_j, \varphi_k >=< f, \varphi_k > \end{aligned} \tag{4.4}$$

y por las propiedades asociativa y distributiva del producto escalar, se llega a que

$$\sum_{j=o}^{n} c_j^* < \varphi_j, \varphi_k >=< f, \varphi_k > \ , \ 0 \leq k \leq n, \tag{4.5}$$

lo cual constituye el sistema de ecuaciones lineales

$$\begin{array}{ll} k = 0: & c_o^* < \varphi_o, \varphi_o > + c_1^* < \varphi_1, \varphi_o > + \cdots + c_n^* < \varphi_n, \varphi_o >=< f, \varphi_o > \\ k = 1: & c_o^* < \varphi_o, \varphi_1 > + c_1^* < \varphi_1, \varphi_1 > + \cdots + c_n^* < \varphi_n, \varphi_1 >=< f, \varphi_1 > \\ \cdots & \cdots \\ k = n: & c_o^* < \varphi_o, \varphi_n > + c_1^* < \varphi_1, \varphi_n > + \cdots + c_n^* < \varphi_n, \varphi_n >=< f, \varphi_n > \end{array}$$

que se puede escribir en forma matricial como

$$Bc^* = h$$

$$B = \begin{bmatrix} < \varphi_o, \varphi_o > & < \varphi_1, \varphi_o > & \cdots & < \varphi_n, \varphi_o > \\ < \varphi_o, \varphi_1 > & < \varphi_1, \varphi_1 > & \cdots & < \varphi_n, \varphi_1 > \\ \cdots & \cdots & \cdots & \cdots \\ < \varphi_o, \varphi_n > & < \varphi_1, \varphi_n > & \cdots & < \varphi_n, \varphi_n > \end{bmatrix} \tag{4.6}$$

$$h = \begin{bmatrix} < f, \varphi_o > \\ < f, \varphi_1 > \\ \cdots \\ < f, \varphi_n > \end{bmatrix}$$

$c = (c_o^*, c_1^*, ..., c_n^*)^T$.

Este sistema de ecuaciones lineales se conoce como sistema de ecuaciones normales (SEN) o de Gauss. La denominación de normales proviene del hecho de la ortogonalidad del residuo a todos los elementos de la base en el espacio de aproximación

$$< f - \widehat{F}, \phi_k >= 0 \quad \textit{equivale a que } f - \widehat{F} \perp \Phi, \tag{4.7}$$

según la generalización del concepto de ortogonalidad, pues la distancia mínima de f al subespacio Φ está dada por la longitud del vector $f - \widehat{F}$, siendo $\widehat{F}$ su proyección ortogonal (hacer gráfico). Además debido a la conmutatividad del producto escalar, la matriz B es simétrica.

La matriz B de los productos escalares del SEN (4.6) es una matriz de Gram, y se puede demostrar que es definida positiva [2] *siempre que las funciones φ_j sean linealmente independientes.*

Luego el sistema tiene solución única c^*, que define la función de mejor aproximación. Para demostrar que cualquier función $F = \sum_j c_j \varphi_j$ con al menos un $c_j \neq c_j^*$ tiene mayor distancia a f que $\widehat{F}$, planteamos la diferencia $f - F = f - \sum_j c_j \varphi_j$. Sumando y restando $\widehat{F}$

$$\begin{aligned} f - F &= (f - \widehat{F}) + (\widehat{F} - \sum_j c_j \varphi_j) \\ &= (f - \widehat{F}) + \sum_j (c_j^* - c_j)\varphi_j \ . \end{aligned}$$

Entonces,

$$\begin{aligned} \|f - F\|_2^2 &=< f - F,\ f - F > \\ &=< f - \widehat{F} + \sum_j (c_j^* - c_j)\varphi_j,\ f - \widehat{F} + \sum_j (c_j^* - c_j)\varphi_j > \\ &=< f - \widehat{F}, f - \widehat{F} > + 2 < \sum_j (c_j^* - c_j)\varphi_j,\ f - \widehat{F} > \\ &+ < \sum_j (c_j^* - c_j)\varphi_j, \sum_j (c_j^* - c_j)\varphi_j > \\ &= \left\| f - \widehat{F} \right\|_2^2 + \left\| \sum_j (c_j^* - c_j)\varphi_j \right\|_2^2 , \end{aligned}$$

pues, teniendo en cuenta (4.7), el sumando que contiene el coeficiente 2 se anula, y como, al menos un $c_j \neq c_j^*$, el segundo sumando en la última expresión es estrictamente positivo, y queda

$$\|f - F\|_2^2 \geq \left\| f - \widehat{F} \right\|_2^2 ,$$

con lo que se completa la demostración.

[2] Una matriz A es definida positiva (semidefinida positiva), $A > 0 (\geq 0)$ si y sólo sí $x^T A x > 0 (x^T A x \geq 0)$ para toda $x \neq 0 (x \in \mathbb{R}^n)$

Lo que se acaba de demostrar es totalmente congruente y equivalente con las exigencias de optimalidad que aseguran la existencia de la solución del problema (4.2), veamos,

$$\left\langle \sum_{k=0}^{n} c_k\varphi_k(x) - f(x), \sum_{k=0}^{n} c_k\varphi_k(x) - f(x) \right\rangle = \sum_{k=0}^{n} c_k \sum_{j=0}^{n} c_j \left\langle \varphi_k(x), \varphi_j(x) \right\rangle \tag{4.8}$$

$$- 2\sum_{0}^{n} c_k Re(\langle f, \varphi_k(x)\rangle) + \langle f, f\rangle \tag{4.9}$$

Como el último sumando no depende de las variables con respecto a las que se está optimizando y asumiendo que $F(x,c)$ y $f(x)$ toman valores reales, pues es suficiente resolver

$$\min_{c\in\mathbb{R}^n} \sum_{k=0}^{n} c_k \sum_{j=0}^{n} c_j \left\langle \varphi_k(x), \varphi_j(x) \right\rangle - 2\sum_{0}^{n} c_k \left\langle f, \varphi_k(x) \right\rangle \tag{4.10}$$

Veamos cuáles son las condiciones de optimalidad para el problema

$$\min_{c\in\mathbb{R}^n} g(c) \tag{4.11}$$

Definición 44

- *c^* es un mínimo global de (4.11) si $\forall\ c \in \mathbb{R}^n$; $g(c) \geq g(c^*)$.*
- *Si existe una vecindad V_{c^*} de c^* tal que $\forall\ c \in V_{c^*} \cap \mathbb{R}^n$; $g(c) \geq g(c^*)$, entonces c^* es un mínimo local de (4.11).*

Definición 45 *La función $g(x)$ es convexa si $\forall x_1, x_2 \in \mathbb{R}^n$, $\alpha \in [0,1]$ se tiene*

$$g(\alpha(x_1) + (1-\alpha)x_2) \leq \alpha g(x_1) + (1-\alpha)g(x_2)$$

Además, si $g(c) \in \mathbb{C}^2$ entonces g es convexa si y solo si $\nabla^2 g(c) \succ 0$.

Teorema 46 *La condición necesaria de mínimo local es como sigue: Si c^* es mínimo local de (4.11) entonces*

- *si $g \in C^1$ entonces $\nabla g(c^*) = 0$.*
- *si $g \in C^2$ entonces $\nabla g(c^*) = 0$ y $\nabla^2 g(c^*)$ es semidefinida positiva.*

Teorema 47 *(Condiciones suficientes). Si $g \in C^2$, $\nabla g(c^*) = 0$ y $\nabla^2 g(c^*)$ es definida positiva entonces c^* es un mínimo local de (4.11).*

Si g es convexa, entonces la condición $\nabla g(c^*) = 0$ es condición suficiente para la existencia del mínimo global. Retomando nuestro problema (4.10), nuestra función $g(c) \in C^\infty$

$$g(c) = \sum_{k=0}^{n} c_k \sum_{j=0}^{n} c_j \left\langle \varphi_k(x), \varphi_j(x) \right\rangle - 2\sum_{0}^{n} c_k \left\langle f, \varphi_k(x) \right\rangle \tag{4.12}$$

y $\nabla g = 2Bc - 2h$, con B la matriz de los productos escalares obtenida más arriba y h el vector de los productos escalares de f con las funciones φ_k que se dijo son linealmente independientes (l.i.), por tanto, $\nabla g(c) = 0$ es equivalente a resolver el sistema de ecuaciones normales lineales $Bc = h$, lo cual se demostró tiene solución única c^*, ya que precisamente al ser las φ_j l.i. la matriz B es definida positiva, con lo cual se obtiene que $\nabla^2 g = 2B$ es definida positiva y por tanto esto implica que la función g es convexa, de ahí que c^* es el único mínimo global.

Aproximación por polinomios. Caso discreto

Si $F \in P_n = \Phi$, entonces

$$F(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n = \sum_{j=0}^{n} c_j x^j,$$

y se dispone de la función f que se quiere aproximar en la forma de una tabla de $N+1$ pares ($N \gg n$, N grande):

x	x_0	x_1	$\cdots$	x_N
f(x)	f_0	f_1	$\cdots$	f_N

En este caso, se pueden tomar como funciones linealmente independientes las potencias de x:

$$\{\varphi_j(x)\}_{j=0}^{n} = \left\{x^j\right\}_{j=0}^{n}.$$

El producto escalar con función de peso $\omega(x) = 1$ está definido por

$$\langle \varphi_j, \varphi_k \rangle = \sum_{i=0}^{N} \varphi_j(x_i)\varphi_k(x_i) = \sum_{i=0}^{N} x_i^j x_i^k = \sum_{i=0}^{N} x_i^{j+k}$$

y

$$\langle f, \varphi_k \rangle = \sum_{i=0}^{N} f(x_i)\varphi_k(x_i) = \sum_{i=0}^{N} f_i x_i^k$$

obteniéndose, por ejemplo,

si $j = k = 0$, $\langle \varphi_0, \varphi_0 \rangle = \sum x_i^{0+0} = \sum 1 = N+1$

si $j = k = 1$ $\langle \varphi_1, \varphi_1 \rangle = \sum x_i^{1+1} = \sum x_i^2$

si $j = 0, k = 1$ $\langle \varphi_0, \varphi_1 \rangle = \sum x_i^{0+1} = \sum x_i$, etc. El sistema (43) de las ecuaciones normales tendrá la forma

$$\begin{bmatrix} N+1 & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{n+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{n+2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \cdots & \sum x_i^{2n} \end{bmatrix} \begin{bmatrix} c_0^* \\ c_1^* \\ c_2^* \\ \vdots \\ c_n^* \end{bmatrix} \tag{4.13}$$

$$= \begin{bmatrix} \sum f_i \\ \sum x_i f_i \\ \sum x_i^2 f_i \\ \vdots \\ \sum x_i^n f_i \end{bmatrix}, \tag{4.14}$$

y bastará resolverlo para hallar el vector c^* de los coeficientes del polinomio $\widehat{F}$ que da la mejor aproximación mínimo cuadrática.

Aspectos computacionales de la aproximación por polinomios

La determinación de la matriz B y el correspondiente término independiente h

$$B = \begin{bmatrix} N+1 & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{n+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \cdots & \sum x_i^{2n} \end{bmatrix},$$

$$h = \begin{bmatrix} \sum f_i \\ \sum x_i f_i \\ \cdots \\ \sum x_i^n f_i \end{bmatrix}$$

requieren el cálculo de todas las sumatorias que estos contienen. Para obtener expresiones que faciliten la automatización de dicho cálculo, denotemos por X la matriz de datos de orden $(N+1) \times (n+1)$ y por f y el vector de $N+1$ componentes como sigue:

$$X = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_N & x_N^2 & \cdots & x_N^n \end{bmatrix}, \qquad f = \begin{bmatrix} f_0 \\ f_1 \\ \cdots \\ f_N \end{bmatrix}.$$

Se puede demostrar que B y f se pueden calcular mediante:

$$B = X^T X \quad \text{y} \quad h = X^T f.$$

La matriz X es una matriz del tipo Vandermonde y se genera fácilmente a partir del vector

$$x = (x_0, x_1, \ldots, x_N)^T.$$

Cuando la función de aproximación F es polinómica, la matriz B es desbalanceada (sus filas y columnas son de orden diverso), lo que ocasiona problemas con la propagación de los errores de redondeo, y si n es grande, resulta ser una matriz mal condicionada.

¿Qué se puede hacer con vista a obtener la solución de las ecuaciones normales con máxima precisión?

Hay dos posibles enfoques:

- utilizar bases $\{\varphi_j\}$ ortogonales.
- no usar las ecuaciones normales.

Sobre esto volveremos después.

Aproximación mínimo cuadrática polinomial en la Estadística

En el caso discreto la aproximación mínimo cuadrática se identifica en la Estadística con el problema llamado de ajuste de datos o determinación de una función de regresión. La regresión lineal es el caso más frecuente en la práctica, que es la aproximación mediante una recta, $\widehat{F}(x) = c_0^* + c_1^* x$ de un conjunto de datos

$$X = \begin{bmatrix} 1 & x_0 \\ 1 & x_1 \\ \cdots & \cdots \\ 1 & x_N \end{bmatrix}, \quad f = \begin{bmatrix} f_0 \\ f_1 \\ \cdots \\ f_N \end{bmatrix}.$$

Aplicando la teoría vista más arriba, se obtiene que el sistema de las ecuaciones normales tiene la forma

$$\begin{bmatrix} N+1 & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} c_0^* \\ c_1^* \end{bmatrix} = \begin{bmatrix} \sum f_i \\ \sum x_i f_i \end{bmatrix}$$

Aproximación mediante una función no lineal, linealizable

Existen casos en los que se propone aproximar los datos con una función que no es lineal con respecto a los parámetros a calcular sin embargo es posible linealizarla. Tal es el caso cuando se aproxima por una función exponencial mínimo cuadrática de la forma $F(x) = c_0 e^{c_1 x}$. La función aproximante puede linealizarse aplicando logaritmos. Aplicando logaritmos se convierte en una recta:

$$\ln F(x) = \ln c_0 + c_1 x,$$

o sea,

$$G(x) = c_0' + c_1 x,$$

donde $G(x) = \ln F(x)$ y $c_0' = \ln c_0$. Con la transformación logarítmica, se ha convertido la función aproximante F que depende en forma no lineal de c_0 y c_1, en la función aproximante G, que depende de c_0' y c_1 linealmente. El problema de aproximación se convierte entonces en hallar $\widehat{G}$ tal que

$$\left\| \ln f - \widehat{G} \right\|_2^2 = \min_G \|\ln f - G\|_2^2 = \sum_{i=0}^N [\ln f_i - (c_0' + c_1 x_i)]^2$$

Está claro que los coeficientes $c_0^* = \exp(c_0'^*)$ y c_1^*, que se obtienen minimizando $\|\ln f - \ln F\|_2^2$, no coinciden con los que se obtendrían minimizando directamente $\|f - F\|_2^2$, los cuales son más difíciles de calcular debido a la no linealidad. Pero en la práctica, por evitar la resolución de un sistema no lineal, se aceptan como tales, pues son bastante cercanos debido a la inyectividad de la transformación logarítmica. Tomando en este caso, también $\{\varphi_j(x)\}_{j=0}^1 = \{1, x\}$ y los datos representados por

$$X = \begin{bmatrix} 1 & x_0 \\ 1 & x_1 \\ \cdots & \cdots \\ 1 & x_N \end{bmatrix}, \quad f = \begin{bmatrix} \ln f_0 \\ \ln f_1 \\ \cdots \\ \ln f_N \end{bmatrix},$$

el SEN será:

$$\begin{bmatrix} N+1 & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} c_0'^* \\ c_1^* \end{bmatrix} = \begin{bmatrix} \sum \ln f_i \\ \sum x_i \ln f_i \end{bmatrix},$$

cuya solución (c_0^*, c_1^*) permite determinar finalmente $c_0^* = e^{c_0'^*}$, y definir la función de aproximación $\widehat{F}(x) = c_0^* e^{c_1^* x}$.

4.1.2. Aproximación lineal múltiple

Si la función empírica f depende linealmente de p variables, $f = f(x_1, x_2, \ldots, x_p)$, y se realizan $N+1$ observaciones, tendremos la tabla siguiente:

observaciones	x_1	x_2	$\cdots$	x_p	f
0	x_{01}	x_{02}	$\cdots$	x_{0p}	f_0
1	x_{11}	x_{12}	$\cdots$	x_{1p}	f_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	x_{i1}	x_{i2}	$\cdots$	x_{ip}	f_i
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
N	x_{N1}	x_{N2}	$\cdots$	x_{Np}	f_N

Las funciones de aproximación tienen la forma

$$F(x_1, x_2, \ldots, x_p) = c_0 + c_1 x_1 + c_2 x_2 + \cdots + c_p x_p,$$

y los vectores $x_1, x_2, \ldots, x_p$ que definen las variables son linealmente independientes, puede considerarse el conjunto de funciones $\{\varphi_j(x)\}_{j=0}^{p} = \{1, x_1, x_2, \ldots, x_p\}$ y los datos representados por:

$$X = \begin{bmatrix} 1 & x_{01} & x_{02} & \cdots & x_{0p} \\ 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Np} \end{bmatrix}, \quad f = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_N \end{bmatrix}, \text{ con lo cual, el SEN tendrá la forma:}$$

$$\begin{bmatrix} N+1 & \sum x_{i1} & \sum x_{i2} & \cdots & \sum x_{ip} \\ \sum x_{i1} & \sum x_{i1}^2 & \sum x_{i1}x_{i2} & \cdots & \sum x_{i1}x_{ip} \\ \sum x_{i2} & \sum x_{x2}x_{i1} & \sum x_{i2}^2 & \cdots & \sum x_{i2}x_{ip} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum x_{ip} & \sum x_{ip}x_{i1} & \sum x_{ip}x_{i2} & \cdots & \sum x_{ip}^2 \end{bmatrix} \begin{bmatrix} c_0^* \\ c_1^* \\ c_2^* \\ \vdots \\ c_p^* \end{bmatrix} = \begin{bmatrix} \sum f_i \\ \sum x_{i1} f_i \\ \sum x_{i2} f_i \\ \vdots \\ \sum x_{ip} f_i \end{bmatrix}.$$

Caso particular 1:*Sistema lineal sobredeterminado.*

La resolución aproximada de un sistema lineal sobredeterminado $Ac = b$, con $A_{n\times m}$, $b_{n\times 1}$, $n > m$, se puede interpretar como la aproximación del vector $b = f \in \mathbb{R}^n$ por la combinación lineal de las columnas de A, donde $a^{(j)}$ es la j-ésima columna de A, que minimice el residuo $r = b - Ac$:

$$\|b - A\widehat{c}\|_2 = \min_c \|b - Ac\|_2 .$$

En este caso, $\{\varphi_j\}_{j=1}^{m} = \left\{a^{(1)}, a^{(2)}, \ldots, a^{(m)}\right\}$, y los datos están representados por:

$$X = A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}, \quad f = b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

El SEN toma entonces la forma , $A^TAc^* = A^Tb$, donde c^* es el vector de los coeficientes desconocidos. En este contexto, se demuestra el siguiente resultado

Teorema 48 *Sean X e Y dos espacios vectoriales de dimensión finita n y m sobre $\mathbb{R}$ y L una transformación lineal representada en dos bases X e Y por la A. Para un vector dado $b \in Y$, el vector $x \in X$ minimiza $\|Ax - b\|_2 \Longleftrightarrow A^TAx = A^Tb$.*

Caso particular 2: Si la función empírica depende en forma no lineal de los coeficientes, pero es linealizable mediante la aplicación de logaritmos, obtenemos el caso lineal múltiple. Por ejemplo, si la función de aproximación es de la forma

$$F(x, y, z) = \alpha \frac{x^\beta y^\gamma}{z^\delta},$$

entonces

$$\ln F = \ln \alpha + \beta \ln x + \gamma \ln y - \delta \ln z,$$

o sea,

$$G = \alpha' + \beta x' + \gamma y' - \delta z',$$

y tenemos una función de aproximación lineal múltiple G. Tomando en este caso $\{\varphi_j\}_{j=o}^3 = \{1, x', y', z'\}$ y los datos representados por:

$$X = \begin{bmatrix} 1 & \ln x_1 & \ln y_1 & -\ln z_1 \\ 1 & \ln x_2 & \ln y_2 & -\ln z_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \ln x_n & \ln y_n & -\ln z_n \end{bmatrix}, \quad f = \begin{bmatrix} \ln f_1 \\ \ln f_1 \\ \vdots \\ \ln f_n \end{bmatrix}$$

se obtiene el SEN para determinar $\widehat{G}$ con coeficientes α', β, γ y δ, que minimiza

$$\|\ln f - G\|_2 = \sum_{i=1}^n [\ln f(x_i, y_i, z_i) - (\alpha' + \beta \ln x_i + \gamma \ln y_i - \delta \ln z_i)]^2.$$

4.2. Aproximación por mínimos cuadrados no lineal

Se define la función S como el cuadrado del error de la aproximación mínimo cuadrática:

$$S = \|f - F\|_2^2 = \sum_{i=0}^N [f(x_i) - F(x_i; c_0, c_1, \ldots, c_n)]^2, \tag{4.15}$$

$$S = S(c_0, c_1, \ldots, c_n).$$

que es una función continua, positiva y diferenciable de los parámetros $c_0, c_1, \ldots, c_n$, por lo menos hasta de segundo orden, $S : \mathbb{R}^n \to \mathbb{R}$. Luego, el problema formulado como sigue: encontrar $c^* \in \mathbb{R}^n$ tal que

$$S(c^*) = \min_{c \in \mathbb{R}^n} S(c)$$

es un problema de optimización sin restricciones. Entonces, se aplican las condiciones de optimalidad vistas anteriormente

$$\nabla S(c) = \left(\frac{\partial S}{\partial c_0}, \frac{\partial S}{\partial c_1}, \ldots, \frac{\partial S}{\partial c_n} \right)^T = \overrightarrow{0}_{\mathbb{R}^{n+1}}$$

Teniendo en cuenta (4.15), y derivando con respecto a los c_j, se obtiene

$$\begin{aligned}\frac{\partial S}{\partial c_j} &= \sum_{i=0}^{N}\left\{\frac{\partial}{\partial c_j}\left([f(x_i)-F(x_i;c_0,\ldots,c_n)]^2\right)\right\}\\ &= -2\sum_{i=0}^{N}\left\{[f(x_i)-F(x_i;c_0,\ldots,c_n]\frac{\partial F}{\partial c_j}\right\},\end{aligned}$$

como habíamos visto la condición necesaria de extremo da lugar al sistema de las ecuaciones normales, que en este caso será no lineal

$$\sum_{i=0}^{N}\left\{[f(x_i)-F(x_i;c_0^*,\ldots,c_n^*]\frac{\partial F}{\partial c_j}\right\}=0,\quad 0\le j\le n.$$

La solución del Sistema de Ecuaciones No Lineales (SEN) es el vector c^* que constituye el único mínimo de S y define la mejor función de aproximación mínimo cuadrática $\widehat{F}$.

Desde el punto de vista computacional, la dificultad fundamental está en la resolución del SEN, que exige el uso de métodos iterativos.

Recíprocamente, si tenemos un sistema (en general, no lineal) de n ecuaciones con m incógnitas:

$$f(x)=0,\quad x=(x_1,x_2,\ldots,x_m)^T,\quad f:\mathbb{R}^m\longrightarrow\mathbb{R}^n,$$

o sea,

$$\begin{aligned}f_1(x)&=0\\ f_2(x)&=0\\ &\cdots\\ f_n(x)&=0\end{aligned}$$

y queremos minimizar el error residual

$$\|f(x)\|_2^2=(f_1(x))^2+(f_2(x))^2+\cdots+(f_n(x))^2,$$

tenemos un problema de optimización sin restricciones.

Ejemplo 49 *Dada la función tabulada f:*

x	x_o	x_1	$\cdots$	x_N
$f(x)$	f_o	f_1	$\cdots$	f_N

Aproximar por una función exponencial mínimo cuadrática de la forma $F(x)=c_0e^{c_1x}$ *, sin linealizar.*

Aplicando la forma general del método de los mínimos cuadrados, definimos

$$S=\sum_{i=0}^{N}[f(x_i)-c_0e^{c_1x_i}]^2.$$

Derivando con respecto a los c_j e igualando a cero, se obtien el SEN:

$$\sum_{i=0}^{N}[\exp(c_1^* x_i) f_i - c_0^* \exp(2c_1^* x_i)] = 0$$
$$\sum_{i=0}^{N} x_i[\exp(c_1^* x_i) f_i - c_0^* \exp(2c_1^* x_i)] = 0.$$

Nótese la no linealidad del SEN con respecto a c_o^* y c_1^*. Su resolución puede realizarse usando el método de Newton, lo que requiere definir una aproximación inicial $c^{(o)}$ que garantice la convergencia del proceso iterativo.

4.2.1. Error de la aproximación mínimo cuadrática

El error de la aproximación mínimo cuadrática está dado por

$$E = \left\| f - \widehat{F} \right\|_2 = \sqrt{< f - \widehat{F}, f - \widehat{F} >}$$

Una vez calculados los coeficientes c_j^* de la mejor aproximación $\widehat{F}$, basta sustituir en la expresión anterior para obtener el error. En el ejemplo sencillo resuelto anteriormente para la aproximación por la mejor recta mínimo cuadrática para el caso discreto, habrá que evaluar $\widehat{F}$ para las mismas abscisas, y calcular después $E = \sqrt{\sum_{i=0}^{3}[f_i - \widehat{F}_i]^2}$:

x	1	2	3	4
$f(x)$	3	5	10	10
$\widehat{F}(x)$	3.1	5.7	8.3	10.9
$(f - \widehat{F})(x)$	-0.1	-0.7	1.7	-0.9
$(f - \widehat{F})^2(x)$	.01	.49	2.89	0.81

de donde, $E = \sqrt{4{,}20} = 2{,}05$.

El valor de E depende de las componentes del vector f, por lo que se puede obtener una aproximación $\widehat{F}$ bastante buena con un valor no necesariamente pequeño para E. De ahí la existencia de otros criterios o formas de medir el error de la aproximación mínimo cuadrática, que en ciertos casos resultan más convenientes. Por ejemplo,

- suma de cuadrados de los errores: $E^2 = \sum_{i=o}^{N}(f_i - \widehat{F}_i)^2$
- desviación cuadrática media: $E/(N+1)$
- varianza: E/N
- desviación típica: $\sqrt{varianza}$
- error relativo: $E/\left\|f\right\|_2$
- otras estadísticas (la mayoría, basadas en E)

Apéndice

Definiciones de norma más usadas

a) Caso continuo : $g \epsilon C_{[a,b]}$

$$\|g\|_2 = \sqrt{\int_a^b g(x)^2 dx} : \quad \text{norma euclideana}$$
$$\|g\|_\infty = \max_{x\epsilon[a,b]} |g(x)| : \quad \text{norma de Chebyshev}$$

Las dos normas representan casos especiales de la norma en L_p :

$$\|g\|_p = \left(\int_a^b |g(x)|^p \, dx \right)^{1/p}.$$

b) Caso discreto, para funciones definidas en una malla o retícula $M = \{x_i\}_{i=1}^m$ constituida por un conjunto finito de puntos; la norma se define como

$$\|g\|_{p,M} = \left(\sum_{i=1}^m |g(x_i|^p \right)^{1/p}.$$

Se dice que esta norma es realmente una *seminorma* si g es continua, ya que en ese caso no se satisface el primero de los requerimientos de la definición para la función g, que puede ser cero en el conjunto M sin ser idénticamente nula.

c) Con función de peso:

Las definiciones de norma pueden generalizarse introduciendo una cierta función positiva $\omega(x)$ para $a < x < b$, llamada función de peso (weight), que en el caso discreto sería un vector $\omega = (\omega(x_1), ..., \omega(x_m))$ y se tiene:

$$\|g\|_{p,\omega} = \left(\int_a^b \omega(x) \, |g(x)|^p \, dx \right)^{1/p}$$
$$\|g\|_{p,\omega,M} = \left(\sum_{i=1}^m \omega(x_i) \, |g(x_i)|^p \right)^{1/p}$$

d) Producto escalar

Dado un espacio L, h y g en L, el producto escalar $< g, h >$ se define como:

$$< g, h >= \int_a^b \omega(x) g(x) h(x) dx, \text{ en el caso continuo}$$
$$< g, h >= \sum_{i=1}^m \omega(x_i) g(x_i) h(x_i), \text{ en el caso discreto.}$$

Para la norma euclidiana se tiene entonces, que

$$\|g\|_2 = \sqrt{< g, g >}.$$

El producto escalar es un número real, y tiene las propiedades siguientes:

$$< g, g >\geq 0 \ \ \forall g, \ \text{y} \ \ < g, g >= 0 \Longrightarrow g = 0$$
$$< g, h >=< h, g > \ : \text{conmutativa}$$
$$< \alpha g, h >= \alpha < g, h > \ : \text{asociativa con respecto a multiplicación por un escalar}$$
$$< g + f, h >=< g, h > + < f, h > \ : \text{distributiva}$$

La introducción del concepto de norma permite generalizar la noción de distancia entre dos elementos de un espacio. El concepto de producto escalar permite, además, hacer la extensión de otras nociones geométricas tales como ángulos y ortogonalidad. Por ejemplo, para g y $h \in L$ se dice que son ortogonales si se cumple que $< g, h >= 0$.

Algunos resultados teóricos generales

Definición 50 *Sea M un espacio métrico, $f : M \to \mathbb{R}$, diremos que f es continua inferiormente en un punto $x^* \in M$ si para toda sucesión $\{x_n\} \subset M$ que converge a x^* se cumple*

$$f(x^*) \leq \lim \inf_{n\to\infty} f(x_n).$$

Definición 51 *Sea M un espacio de Banach, $\varphi \in (M, \mathbb{R})$ un funcional, $B \subseteq M$ convexo. Se dice que φ es fuertemente convexa en B si: $\frac{1}{2}\varphi(x) + \frac{1}{2}\varphi(y) - \varphi(\frac{x+y}{2}) \geq \gamma \left\|x - y\right\|^2$, para $x, y \in B$, $\gamma > 0$.*

Teorema 52 *Sea M un espacio de Banach, $\varphi \in (M, \mathbb{R})$ un funcional continuo inferiormente, acotado inferiormente y fuertemente convexo sobre el conjunto cerrado (convexo) $B \subseteq M$. Entonces φ alcanza su valor mínimo en $u \in B$ determinado de forma única.*

Teorema 53 *Sea H un espacio de Hilbert, $B \subseteq H$ acotado, convexo y cerrado. Entonces todo funcional $f \in H^*$(con H^*se denota el espacio dual de H que es el conjunto de las aplicaciones lineales y continuas de $H \to \mathbb{R}$ y se denota también por $\mathcal{L}(H, \mathbb{R})$) alcanza su mínimo en B.*

4.3. Aproximación mínimo cuadrática con funciones base ortogonales

En las secciones precedentes se estudió el caso de aproximación mínimo cuadrática lineal en el espacio de las funciones polinómicas. Al aproximar por un polinomio de grado n la matriz de Gram B del Sistema de Ecuaciones Normales resultó ser en general, mal condicionada, característica que se acentúa en la medida en que mayor sea n, pues es consecuencia de lo desbalanceada que es la matriz B :

$$B = \begin{bmatrix} N+1 & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{n+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \cdots & \sum x_i^{2n} \end{bmatrix}.$$

Para reducir el mal condicionamiento, y aumentar con ello la precisión de la solución $c^\star$, se deben tomar ciertas medidas, entre las que destacan:

- tomar n lo menor posible, siempre que el comportamiento de F sea semejante a f
- elegir cuidadosamente el conjunto de funciones base $\{\varphi_j\}_{j=o}^{n}$, para que además de linealmente independiente, sea ortogonal
- no generar las ecuaciones normales $X^T X c = X^T f$, sino tratar directamente la minimización de $\|f - Xc\|$ por la vía de descomponer la matriz de datos X en factores

4.3.1. Funciones base ortogonales en el espacio de los polinomios

Definición 54 *Las funciones φ_j son ortogonales (ortonormales) con respecto a los puntos $x_o, x_1, \ldots, x_N$ y función de peso $\omega(x) \equiv 1$, si se cumple :*

$$\langle \varphi_j, \varphi_k \rangle = \sum_{i=o}^{N} \varphi_j(x_i)\varphi_k(x_i) = 0,\ j \neq k \tag{4.16}$$

$$\langle \varphi_j, \varphi_k \rangle = \delta_{jk} = \left\{ \begin{array}{lll} 0, & si & j \neq k \\ 1, & si & j = k \end{array} \right\})$$

Nos proponemos ahora, con el uso del conjunto $\{\varphi_j\}_{j=o}^{n}$ de funciones ortogonales con respecto a los puntos x_i , disminuir el mal condicionamiento de las ecuaciones normales cuando F es polinómica:

$$F(x) = \sum_{j=o}^{n} c_j x^j \tag{4.17}$$

Las funciones x^j en general, no son ortogonales en los puntos x_i. Consideremos un conjunto de polinomios $p_j(x)$ ortogonales con respecto a los puntos x_i. Entonces, $< p_j(x), p_k(x) >= 0$ para $j \neq k$, y el sistema de las ecuaciones normales se convierte en

$$\left[\begin{array}{cccc} < p_o, p_o > & 0 & \cdots & 0 \\ 0 & < p_1, p_1 > & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & < p_n, p_n > \end{array} \right] \left[\begin{array}{c} c_o^\star \\ c_1^\star \\ \vdots \\ c_n^\star \end{array} \right] = \left[\begin{array}{c} < f, p_o > \\ < f, p_1 > \\ \vdots \\ < f, p_n > \end{array} \right] \tag{4.18}$$

con matriz diagonal, mejor condicionada que la que se obtiene con $\varphi_j = x^j$.

La solución de (4.18) está dada entonces por

$$c_j^* = \frac{< f, p_j >}{< p_j, p_j >} \ , \quad 0 \leq j \leq n \tag{4.19}$$

lo cual requiere sólo la realización de un cociente para obtener cada coeficiente c_j^*, que es independiente de los demás. Si se exige además que $< p_j, p_j >= 1$, es decir, que las funciones $p_j(x)$ sean ortonormales, entonces $c_j^* =< f, p_j >$, y se evita el cáculo del cociente.

Ejemplo:

Dada la función f por la siguiente tabla

x	998	999	1000	1001	1002
f(x)	3.7	4.2	5.1	5.9	6.2

Consideremos la familia de rectas $\{\varphi_j(x)\}_{j=o}^{1}$, linealmente independientes y ortogonales respectivamente.

a) Para el conjunto $\varphi = \{1, x\}$ formado por dos funciones linealmente independientes, se tiene

$$\begin{bmatrix} <\varphi_o,\varphi_o> & <\varphi_o,\varphi_1> \\ <\varphi_1,\varphi_o> & <\varphi_1,\varphi_1> \end{bmatrix} \begin{bmatrix} c_o^* \\ c_1^* \end{bmatrix} = \begin{bmatrix} <f,\varphi_o> \\ <f,\varphi_1> \end{bmatrix} \tag{4.20}$$

$$\begin{bmatrix} 5 & 5000 \\ 5000 & 5000010 \end{bmatrix} \begin{bmatrix} c_o^* \\ c_1^* \end{bmatrix} = \begin{bmatrix} 25{,}1 \\ 25131{,}8 \end{bmatrix} \tag{4.21}$$

resolviendo se obtiene $c^* = \begin{bmatrix} -3174{,}98 \\ 3{,}18 \end{bmatrix}$ y la expresión para la función aproximante es $\widehat{F}(x) = -3174{,}98 + 3{,}18x$ con error $E = 7{,}94$.

En este caso, la matriz B es muy mal condicionada $(cond_2(B) \approx 5 \times 10^{11})$, y si se consideran sólo 4 cifras decimales, entonces es singular:

$det(B) = \begin{bmatrix} 5 & 5000 \\ 5000 & 5000 \times 10^3 \end{bmatrix} = 0$

b) Para el conjunto $\varphi = \{p_o(x), p_1(x)\} = \{1, x - \overline{x}\} = \{1, x - 1000\}$ formado por funciones ortogonales, las ecuaciones normales son

$$\begin{bmatrix} <p_o,p_o> & 0 \\ 0 & <p_1,p_1> \end{bmatrix} \begin{bmatrix} c_o^* \\ c_1^* \end{bmatrix} = \begin{bmatrix} <f,p_o> \\ <f,p_1> \end{bmatrix} \tag{4.22}$$

$$\begin{bmatrix} 5 & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} c_o^* \\ c_1^* \end{bmatrix} = \begin{bmatrix} 25{,}1 \\ 6{,}7 \end{bmatrix} \tag{4.23}$$

de donde se obtiene $c^* = \begin{bmatrix} 5{,}02 \\ 0{,}67 \end{bmatrix}$ siendo la expresión para la función aproximante $\widehat{F}(x) = 5{,}02 + 0{,}67(x - 1000)$ con error $E = 0{,}3146$.

La matriz B ahora es bien condicionada $(cond_2(B) = \sqrt{\rho(B^TB).\rho((B^TB)^{-1})} = \sqrt{10^2/5^2} = 2\ )$.

Este ejemplo pone en evidencia la mejoría que se logra en la función aproximante cuando se usa una base ortogonal.

4.3.2. Generación de polinomios ortogonales

Definición 55 *Se dice que $p_o(x), p_1(x), p_2(x), \ldots$ es una sucesión de polinomios ortogonales siempre que*

$$\langle p_j(x), p_k(x)\rangle = 0, j \neq k \tag{4.24}$$

siendo cada $p_j(x)$ un polinomio de grado j exactamente , es decir, $\forall j$, $p_j(x) = c_j x^j +$ un polinomio de grado menor que j, con $c_j \neq 0$.

Por ejemplo, las funciones

$$p_o(x) = 1, p_1(x) = x, p_2(x) = 3x^2 - 1, \omega(x) = 1, \tag{4.25}$$

constituyen una sucesión de tres polinomios ortogonales si el producto escalar se define por

$$< p_j, p_k >= \int_{-1}^{1} p_j(x).p_k(x)dx \tag{4.26}$$

pues $< p_o, p_1 >=< p_o, p_2 >=< p_1, p_2 >= 0$,

$< p_o, p_o >= 2, \ < p_1, p_1 >= 2/3, \ < p_2, p_2 >= 8/5.$

Proposición 56 *1. Si $p(x)$ es un polinomio de grado $\leq n$, entonces se puede escribir como*

$$p(x) = \sum_{j=o}^{n} c_j.p_j(x) \tag{4.27}$$

En este caso (4.18), la matriz del sistema de ecuaciones normales es diagonal y los coeficientes c_j^* de la aproximación mínimo cuadrática vienen dados por (4.19). Como el cálculo de cada coeficiente es independiente de los demás, si se aumenta el valor de n, no es necesario recalcular los que ya se tenían. Además, como el sistema es ahora mucho más sencillo de resolver, el error de redondeo se reduce considerablemente.

2. Los polinomios ortogonales satisfacen la relación recurrente de Forsythe (fórmula de tres elementos):

$$p_{j+1}(x) = \alpha_j(x - \beta_j).p_j(x) - \gamma_j.p_{j-1}(x), j = 0, 1, 2, \ldots \tag{4.28}$$

$$p_o(x) = 1, \quad p_{-1}(x) = 0,$$

donde $\alpha_j = \frac{A_{j+1}}{A_j}$, $\beta_j = \frac{<xp_j,p_j>}{<p_j,p_j>}$, y $\gamma_j = \frac{\alpha_j<p_j,p_j>}{\alpha_{j-1}<p_{j-1},p_{j-1}>}$ para $j > 0$.

Esta propiedad se puede utilizar para generar sucesiones de polinomios ortogonales, siempre que $< p_j, p_j >\neq 0$, tanto en el caso continuo como en el discreto. En este último, tan importante en las aplicaciones, la sucesión de polinomios ortogonales debe ser generada **a la medida** para cada conjunto de abscisas x_i, ya que los coeficientes β y γ dependen de ellas, y resulta entonces de utilidad la relación recurrente de Forsythe. En general se escogen A_{j+1} y A_j , o sea los números α_j, de modo que la sucesión resultante sea particularmente simple en algún sentido.

Demostración

Supongamos que $p_o(x), p_1(x), \ldots, p_j(x)$ son ortogonales con respecto a la norma 2 en el caso discreto. Supongamos $\alpha_j = 1$ para todo j. Para calcular los coeficientes β_j y γ_j, se impone la condición de ortogonalidad: $< p_k, p_j >= \sum_{i=o}^{N} p_k(x_i)p_j(x_i) = 0$ si $k \neq j$. Multiplicando la fórmula de recurrencia por $p_k(x), (k \leq j)$, y sumando con respecto a los x_i se obtiene :

$$\begin{aligned} \sum_{i=o}^{N} p_k(x_i).p_j(x_i) &= \sum_{i=o}^{N} p_k(x_i).(x - \beta_j).p_j(x_i) - \sum_{i=o}^{N} p_k(x_i).\gamma_j.p_{j-i}(x_i) \\ 0 &= \sum (x_i - \beta_j).p_k(x_i).p_j(x_i) - \gamma_j \sum p_k(x_i).p_{j-1}(x_i) \end{aligned}$$

intervalo $[a,b]$	$\omega(x)$	expresión recurrente	polinomios de
$[-1,1]$	1	$P_{j+1}(x) = \frac{2j+1}{j+1}P_j(x) - \frac{j}{j+1}P_{j-1}(x)$	Legendre
$(-1,1)$	$\frac{1}{\sqrt{1-x^2}}$	$T_{j+1}(x) = 2xT_j(x) - T_{j-1}(x)$	Tchebyshev
$[0,\infty)$	e^{-x}	$L_{j+1}(x) = \sum_{k=o}^{j+1} \left(\begin{array}{c} j+1 \\ k \end{array} \right) (-x)^k/k!$	Laguerre
$(-\infty,\infty)$	e^{-x^2}	$H_{j+1}(x) = 2xH_j(x) - 2jH_{j-1}(x)$	Hermite

Cuadro 4.1: Expresiones para polinomios

Para $k = j$,

$$0 = \sum x_i.p_j^2(x_i) - \beta_j \sum p_j^2(x_i) - \gamma_j \sum p_j(x_i).p_{j-1}(x_i)$$

luego, como p_j y p_{j-1} son ortogonales,

$$\beta_j = \frac{\sum x_i.p_j^2(x_i)}{\sum p_j^2(x_i)} = \frac{< xp_j, p_j >}{< p_j, p_j >}.$$

Para $k = j-1$,

$$0 = \sum x_i.p_{j-1}(x_i).p_j(x_i) - \beta_j \sum p_{j-1}(x_i).p_j(x_i) - \gamma_j \sum p_{j-1}^2(x_i)$$

por tanto, análogamente

$$\gamma_j = \frac{\sum x_i.p_{j-1}(x_i).p_j(x_i)}{\sum p_{j-1}^2(x_i)} = \frac{< x.p_{j-1}, p_j >}{< p_{j-1}, p_{j-1} >}$$

y se puede demostrar que la expresión de γ_j es equivalente a

$$\gamma_j = \frac{< p_j, p_j >}{< p_{j-i}, p_{j-1} >},$$

que es más conveniente para el cálculo.

En el caso continuo se dispone en la literatura de tablas de polinomios ortogonales construidas previamente, como se puede ver en el cuadro (4.1): (Kie, p156 y Conte, p252)

De las expresiones para las funciones de peso $\omega(x)$ se puede apreciar que la primera no pondera (ya que es constante en todo el intervalo), la segunda da mayor peso cerca de ambos extremos del intervalo, la tercera en el extremo izquierdo, y la cuarta en el centro del intervalo. En el cuadro (4.2) se muestran los primeros elementos de dichas sucesiones de polinomios,

Si F no es de forma polinómica, lo cual se presenta con mucha frecuencia en la práctica, existen otras maneras de construir una sucesión de funciones ortogonales, como por ejemplo, el algoritmo de Gram-Schmidt, y mejor todavía, el modificado de Gram-Schmidt, que goza de la estabilidad numérica que no tiene el primero.

4.3.3. Resolución sin ecuaciones normales

Cuando la matriz X^TX del SEN es mal condicionada, se puede evadir la formación de X^TX y X^Tf , y hallar la solución aproximada que minimiza el cuadrado de la norma euclideana del residuo

k	Legendre	Chebyshev	Laguerre	Hermite
0	1	1	1	1
1	x	x	1-x	2x
2	$\frac{1}{2}(3x^2-1)$	$2x^2-1$	$2-4x+x^2$	$4x^2-2$
3	$\frac{1}{2}(5x^3-3x)$	$4x^3-3x$	$6-18x+9x^2-x^3$	$8x^3-12x$
4	$\frac{1}{8}(35x^4-30x^2+3)$	$8x^4-8x^2+1$	$24-96x+72x^2-16x^3+x^4$	$16x^4-48x^2+12$

Cuadro 4.2: Primeros elementos para la sucesión de polinomios

$\|f-Xc\|_2^2$, en lugar de la del residuo $\left\|X^Tf-X^TXc\right\|_2^2$. El objetivo de este tratamiento es reducir el mal condicionamiento de la matriz $B=X^TX$, ya que $cond_2(X)=\sqrt{cond_2(X^TX)}$. Obtendremos dos descomposiciones ortogonales de la matriz de datos X: la ortogonal de Householder y la descomposición en valores singulares (SVD).

a) Aplicación de la transformación ortogonal de Householder

Esta transformación ortogonal H conserva la norma euclideana, ya que

$$\|H(f-Xc)\|_2^2=\|f-Xc\|_2^2$$

y posee, por tanto, buenas cualidades numéricas. El problema de calcular c^* se reduce entonces a resolver el sistema transformado sobredeterminado $HXc=Hf$, donde se sustituye X por su correspondiente descomposición del tipo QR con $Q=H^T$ ortogonal de Householder ($H^T=H^{-1}$) y R triangular superior, $X=H^TR$. Supondremos que X es de rango completo (igual al número de sus columnas), entonces el problema se convierte en

$$\begin{aligned} H.H^TRc &= Hf \\ Rc &= \widetilde{f} \\ \begin{bmatrix} \widetilde{R} \\ 0 \end{bmatrix} c &= \begin{bmatrix} \widetilde{f}_n \\ \widetilde{f}_{N-n} \end{bmatrix}, \end{aligned}$$

que da la mejor aproximación mínimo cuadrática de norma mínima c^* como solución del subsistema

$$\widetilde{R}c^*=\widetilde{f}_n, con\ error\left\|\widetilde{f}_{N-n}\right\|_2=\sqrt{\sum_{i=n+1}^{N}\widetilde{f}_i^2}. \tag{4.29}$$

b) Aplicación de la descomposición singular (SVD)

La descomposición singular de la matriz X , de N filas y n columnas ($N>>n$), y columnas linealmente independientes (o sea de rango completo igual a n), tiene la forma:

$$X=USV^T,\ \text{con}\ U_{N\times N},S_{N\times n},V_{n\times n} \tag{4.30}$$

donde U y V son matrices ortogonales y

$$S = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \cdots & \ddots & \cdots \\ 0 & \cdots & \sigma_n \\ 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} \widetilde{S} \\ 0 \end{bmatrix}$$

es rectangular con un bloque diagonal $n \times n$ que contiene los valores singulares $\sigma_i > 0$ y un bloque de ceros $(N-n) \times n$. Los valores singulares guardan con los valores propios λ_i de las matrices $X^T X$ y XX^T la relación $\sigma_i^2 = \lambda_i$, $(1 \leq i \leq n)$.

En este caso, la obtención de c^*que minimiza $\|f - Xc\|_2^2$ se reduce a resolver el sistema sobredeterminado $Xc = f$ que, teniendo en cuenta la factorización 4.30, se puede escribir como $USV^T c = f$, y por la ortogonalidad de U, entonces $SV^T c = U^T f$. Denotando $d = V^T c \in \Re^n$, $\widetilde{f} = U^T f \in \Re^N$, y de acuerdo con la forma de S, se obtiene:

$$\begin{bmatrix} \sigma_1 d_1 \\ \vdots \\ \sigma_n d_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \widetilde{f}_1 \\ \vdots \\ \widetilde{f}_n \\ \widetilde{f}_{n+1} \\ \vdots \\ \widetilde{f}_N \end{bmatrix} \quad \curvearrowright \quad \begin{bmatrix} \widetilde{S} \\ 0 \end{bmatrix} d = \begin{bmatrix} \widetilde{\widetilde{f}}_n \\ \widetilde{\widetilde{f}}_{N-n} \end{bmatrix}.$$

De aquí, $\widetilde{S}d = \widetilde{\widetilde{f}}_n \quad \curvearrowright \quad d_i = \frac{\widetilde{f}_i}{\sigma_i}$, $(1 \leq i \leq n)$, luego después de calculado el vector d, la mejor aproximación mínimo cuadrática de norma mínima c^* se obtiene , teniendo en cuenta la ortogonalidad de V, por

$$c^* = Vd, \text{ con error } \left\| \widetilde{\widetilde{f}}_{N-n} \right\|_2 = \sqrt{\sum_{i=n+1}^{N} \widetilde{f}_i^2}. \tag{4.31}$$

Observación 57 *Las expresiones anteriores (4.29) y (4.31) para el error de la solución mínimo cuadrática de norma mínima c^* también pueden obtenerse en términos de la pseudoinversa X^+, considerando las factorizaciones mencionadas:*

$$\begin{aligned} c^* &= X^+ f \longrightarrow \left(H^T \begin{bmatrix} \widetilde{R} \\ 0 \end{bmatrix} \right)^+ \\ {}^+f &= \begin{bmatrix} \widetilde{R} & 0 \end{bmatrix}^+ \\ Hf &= \widetilde{R}^{-1} \widetilde{\widetilde{f}}_n \end{aligned}$$

para la de Householder, y

$$\begin{aligned} c^* &= X^+ f \longrightarrow \left(U \left[\begin{array}{c} \widetilde{S} \\ 0 \end{array} \right] V^T \right)^+ \\ {}^+f &= V \left[\begin{array}{cc} \widetilde{S} & 0 \end{array} \right]^+ \\ U^T f &= V \widetilde{S}^{-1} \widetilde{\widetilde{f}}_n \end{aligned}$$

para la SVD.

La pseudoinversa X^+ (de Moore y Penrose) cumple las propiedades: $XX^+X = X$, $X^+XX^+ = X^+$, XX^+ y X^+X simétricas, siendo la inversa usual X^{-1} un caso particular de la misma cuando X es cuadrada y no singular.

Ejemplo:

Aplicando la transformación de Householder al mismo ejemplo anterior, a partir de los datos:

$$X = \left[\begin{array}{cc} 1 & 998 \\ 1 & 999 \\ 1 & 1000 \\ 1 & 1001 \\ 1 & 1002 \end{array} \right], \qquad f = \left[\begin{array}{c} 3{,}7 \\ 4{,}2 \\ 5{,}1 \\ 5{,}9 \\ 6{,}2 \end{array} \right]$$

se obtienen,

$$HX = \left[\begin{array}{c} \widetilde{R} \\ 0 \end{array} \right] = \left[\begin{array}{cc} -{,}0022 & -2{,}2361 \\ 0 & 0{,}0032 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \quad \text{y} \quad Hf = \widetilde{f} = \left[\begin{array}{c} -11{,}2251 \\ 2{,}1187 \\ 0{,}0466 \\ 0{,}1325 \\ -0{,}2815 \end{array} \right]$$

de donde,

$$c^* = \left[\begin{array}{c} \widetilde{R} \\ 0 \end{array} \right]^+ \widetilde{f} = \left[\begin{array}{c} -664{,}98 \\ 0{,}67 \end{array} \right] \quad y \quad \widehat{F}(x) = -664{,}98 + 0{,}67x,$$

siendo el error, $\left\| f - \widehat{F} \right\|_2 = \| f - Xc^* \| = 0{,}3146$.

Por otra parte aplicando la descomposición singular, $X = USV^T$, por lo que $Xc = f$ se sustituye por $USV^Tc = f$, y premultiplicando por U^T,

$$\left[\begin{array}{c} \widetilde{S} \\ 0 \end{array} \right] V^T c = \left[\begin{array}{cc} 2236{,}1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] V^T c \quad \text{y} \quad U^T f = \widetilde{f} = \left[\begin{array}{c} 11{,}2280 \\ -2{,}1028 \\ 0{,}0466 \\ 0{,}1325 \\ -0{,}2815 \end{array} \right]$$

luego,

$$c^* = V \begin{bmatrix} \widetilde{S} \\ 0 \end{bmatrix}^+ \widetilde{f} = \begin{bmatrix} -664{,}98 \\ 0{,}67 \end{bmatrix}$$

con error $\|f - Xc^*\|_2 = 0{,}3146$.

Como puede observarse, por las tres vías mencionadas que usan una base de polinomios ortogonales o descomposiciones ortogonales de la matriz de datos, los resultados son iguales y el error cuadrático es mucho menor.

4.4. Funciones base ortogonales en el espacio $L^2[0,2\pi]$

Consideremos ahora que el conjunto de datos que se desea modelar presenta un comportamiento cíclico como por ejemplo, los pulsos en las líneas de telecomunicaciones, el movimiento de los glaciales, la producción de CO_2 en la atmósfera de la tierra, la señal de un electrocardiograma, entre otros. Es lógico entonces pensar en usar funciones que también muestren un comportamiento periódico. Si consideramos el sistema ortonormal $\{\varphi_k\}_{k\in\mathbb{N}} = \left\{1, e^{\pm ikx}\right\}_{k\in\mathbb{N}}$ en el intervalo $[-\pi,\pi]$ estamos simplemente ante la **serie trigonométrica de Fourier** s, que aproxima a la extensión 2π periódica de $f(x) \in L^2[-\pi,\pi]$(espacio de las funciones de cuadrado integrables)[3]

$$s := \sum_{k=-\infty}^{\infty} c_k e^{ikx},\ c_k(f) := \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-ikx}dx \tag{4.32}$$

Si consideramos que para funciones complejas, el producto escalar en el caso continuo se define como

$$\langle g,h\rangle = \frac{1}{2\pi}\int_0^{2\pi} g(x)\,\overline{h(x)}dx$$

y en el caso discreto como

$$\langle g,h\rangle = \frac{1}{N}\sum_{l=0}^{N-1} g(x_l)\,\overline{h(x_l)}$$

entonces dejamos de ejercicio al lector demostrar que tomando como base del espacio Φ el conjunto de funciones $\{\varphi_j(x)\}_{j=-n}^{n} = \{1, e^{\pm ix}, e^{\pm i2x}, ..., e^{\pm nx}\}$, es una base ortonormal con respecto tanto al producto escalar continuo como al discreto.

Para aproximar una función $f(x)$ por la serie de Fourier es necesario que la serie de Fourier sea convergente a la función $f(x)$, en el caso en que no se tiene convergencia de la serie se consideran los polinomios trigonométricos.

4.4.1. Polinomio trigonométrico de Fourier

Definición 58 *Un polinomio trigonométrico de orden n es cualquier función de la forma*

$$p(x) = \sum_{j=-n}^{n} c_j e^{ijx} \tag{4.33}$$

donde c_j son constantes reales o complejas.

[3]es suficiente asumir que f es una función continua por tramos en el intervalo $[0,2\pi]$, entonces $f \in L^2[0,2\pi]$

Tal polinomio trigonométrico es 2π periódico y no es más que la suma parcial $n-$ésima de la serie de Fourier (4.32). Si retomamos lo visto en la teoría general para la aproximación por mínimos cuadrados, la expresión para los coeficientes de Fourier $\widehat{f}(j) = c_j$ del polinomio trigonométrico de orden n,

$$p(x) = \sum_{j=-n}^{n} c_j e^{ijx}$$

de la función $f(x)$ se obtienen resolviendo el sistema de ecuaciones normales $Bc = h$, que tendrá la forma:

$$\begin{bmatrix} < \varphi_o, \varphi_o > & \dots & 0 & & \\ 0 & \dots & 0 & & \\ & \ddots & & & \\ & & < \varphi_n, \varphi_n > & & \\ & & & \ddots & \\ & & & & < \varphi_{-1}, \varphi_{-1} > \end{bmatrix} \begin{bmatrix} c_0^* \\ c_1^* \\ \vdots \\ c_n^* \\ \vdots \\ c_{-1}^* \end{bmatrix} = \begin{bmatrix} < f, 1 > \\ < f, e^{ix} > \\ \vdots \\ < f, e^{inx} > \\ \vdots \\ < f, e^{-ix} > \end{bmatrix}$$

Teniendo en cuenta que el sistema $\{\varphi_j(x)\}_{j=-n}^{n} = \{1, e^{\pm ix}, e^{\pm i2x}, ..., e^{\pm nx}\}$ es ortonormal , la matriz B, además de ser diagonal, es la identidad, luego los coeficientes c_j^* se obtienen directamente como:

$$c_j^* = \widehat{f}(j) = \langle f, \varphi_j \rangle = \left\langle f, e^{ijx} \right\rangle = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-ijx} dx \tag{4.34}$$

$$c_j^* = \widehat{f_N}(j) = \langle f, \varphi_j \rangle = \left\langle f, e^{ijx} \right\rangle_N = \frac{1}{N} \sum_{l=0}^{N-1} f(x_l) e^{-ijx_l} \tag{4.35}$$

para el caso continuo y discreto respectivamente; N es el número de subdivisiones del intervalo $[0, 2\pi]$ y $x_l = \frac{2\pi}{N} l$, los puntos muestrales. La notación $\widehat{f}(j)$ y $\widehat{f_N}(j)$ para los coeficientes de Fourier obedece a que las expresiones (4.34) y (4.35) pueden ser interpretadas como la transformada continua de Fourier y la transformada discreta de Fourier respectivamente.

Teniendo en cuenta la fórmula de Euler $e^{ix} = \cos x + isenx$, la serie de Fourier en (4.32) se puede escribir como

$$s := \frac{a_0}{2} + \sum_{j=1}^{\infty} [a_j \cos jx + b_j senjx] \tag{4.36}$$

con

$$\begin{aligned} a_j &= (c_j + c_{-j}) \\ b_j &= i(c_j - c_{-j}) \end{aligned}$$

Considerando que $\cos(-jx) = \cos(jx)$ y $sen(-jx) = -sen(jx)$ se tiene:

$$\begin{aligned} \sum_{j=-\infty}^{\infty} c_j e^{ijx} &= \sum_{j=-\infty}^{\infty} c_j [\cos jx + isenjx] \\ &= c_0 + \sum_{j=1}^{\infty} (c_j + c_{-j}) \cos(jx) + i(c_j - c_{-j}) sen(jx) \end{aligned} \tag{4.37}$$

Entonces para obtener (4.36) de (4.32) hacer

$$a_j = (c_j + c_{-j})\,; \quad b_j = i\,(c_j - c_{-j}) \tag{4.38}$$

o de manera inversa

$$c_j = \frac{a_j - ib_j}{2}; \quad c_{-j} = \frac{a_j + ib_j}{2}. \tag{4.39}$$

El polinomio triginométrico se puede escribir como

$$p\,(x) = \frac{a_0}{2} + \sum_{j=1}^{n} \left[a_j \cos jx + b_j sen jx\right]. \tag{4.40}$$

Teorema 59 *Sea $f \in V = L^2([-\pi,\pi])$ y $\{\varphi_k\}_{k=1}^n = \{1, coskx, senkx\}_{k=1}^n$ un sistema ortonormal de dimensión finita que genera a V_n,*

$$p(x) = \frac{a_0}{2} + \sum_{k=1}^{n} a_k coskx + b_k senkx$$

con a_k, b_k coeficientes de Fourier de f. Entonces $p(x)$ es el elemento en V_n que está más cercano a f en la norma L^2, esto es

$$\|f - p(x)\|_{L^2} = \min_{g \in V_n} \|f - g\|_{L^2}\,.$$

(Es decir la suma parcial $\sum_{j=-n}^{n} \widehat{f}\,(j)\, e^{ijx}$ de la serie de Fourier para $f\,(x)$ es la mejor aproximación a $f\,(x)$ mediante polinomios trigonométricos de orden n con respecto a la norma $\|g\|_2 = \left[\frac{1}{2\pi}\int_0^{2\pi} \left|g\,(x)^2\,dx\right|\right]^{\frac{1}{2}}$.)

Demostración 60

$$\begin{aligned} \|f-u\|_2^2 &= \left(f - \sum_{j=-n}^{j=n} c_j\varphi_j(x), f - \sum_{j=-n}^{j=n} c_j\varphi_j(x)\right) \\ &= \|f\|_2^2 - 2\sum_{j=-n}^{j=n} c_j(f,\varphi_j(x)) + \sum_{k=-n}^{k=n}\sum_{j=-n}^{j=n} c_j c_k(\varphi_j(x),\varphi_k(x)) \\ &= \|f\|_2^2 - 2\sum_{j=-n}^{j=n} c_j(f,\varphi_j(x)) + \sum_{j=-n}^{j=n} c_j^2 \end{aligned} \tag{4.41}$$

completando cuadrados entre los dos últimos sumandos

$$\|f\|_2^2 - 2\sum_{j=-n}^{j=n} c_j(f,\varphi_j(x)) + \sum_{j=-n}^{j=n} c_j^2 = \|f\|_2^2 + \sum_{j=-n}^{j=n} \left[c_j - (f,\varphi_j(x))\right]^2 - \sum_{j=-n}^{j=n} (f,\varphi_j(x))^2 \tag{4.42}$$

el segundo sumando se hace cero precisamente para $c_j = (f,\varphi_j)$, de ahí que el menor valor se alcance para

$$v = (f,\varphi_j)\varphi_j$$

además por propiedades de la norma

$$0 \leq \|f - v\|_2^2 = \|f\|_2^2 - \sum_{j=-n}^{j=n} (f, \varphi_j(x))^2$$

de lo anterior se obtiene la desigualdad de Bessel

$$\sum_{j=-n}^{j=n} (f, \varphi_j(x))^2 \leq \|f\|_2^2$$

y si $\|f - v\|_2^2 = 0$ *entonces* $\|f\|_2^2 = \sum_{j=-n}^{j=n} (f, \varphi_j(x))^2$*, que se conoce como la igualdad de Parseval.*

Según el teorema de Dirichlet esta serie converge a $f(x)$, en los puntos de continuidad de la función y a la semisuma de los límites laterales en los puntos de discontinuidad, siendo $f(x)$ periódica con período 2π con $f(x)$ y $f'(x)$ seccionalmente continuas en el intervalo $[-\pi, \pi]$.

Teorema 61 *Supongamos que*

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k cos(kx) + b_k sen(kx)$$

con $\sum_{k=1}^{\infty} |a_k| + |b_k| < \infty$*. Entonces la serie trigonométrica de Fourier converge uniforme y absolutamente a la función* $f(x)$*.*

Demostración 62 *Se tiene que*

$$|a_k cos(kx) + b_k sen(kx)| \leq |a_k| + |b_k|$$

lo cual implica que la tasa de convergencia de la serie de Fourier de f *en cualquier punto* x *está gobernada por la tasa de convergencia de* $\sum_k |a_k| + |b_k|$*, entonces para* $S_N(x) = \frac{a_0}{2} + \sum_{k=1}^{N} a_k cos(kx) + b_k sen(kx)$*,*

$$f(x) - S_N(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k cos(kx) + b_k sen(kx) - (\frac{a_0}{2} + \sum_{k=1}^{N} a_k cos(kx) + b_k sen(kx)) \tag{4.43}$$

$$= \sum_{k=N+1}^{\infty} a_k cos(kx) + b_k sen(kx) \tag{4.44}$$

de donde

$$|f(x) - S_N(x)| \leq \sum_{k=N+1}^{\infty} |a_k| + |b_k|, \text{ uniformemente } \forall x$$

entonces como la parte derecha de la desigualdad representa el resto de una serie convergente, tenderá a cero, por lo que el resto de la parte izquierda se puede hacer tan pequeño como se quiera eligiendo N *lo suficientemente grande. Entonces dado* $\epsilon > 0$*, existe* $N_0 > 0(entero)$ *t. q. si* $N > N_0$ *entonces* $\sum_{k=N+1}^{\infty} |a_k| + |b_k| < \epsilon \Rightarrow |f(x) - S_N(x)| \leq \epsilon$*,* $\forall x$*, y como* $N > N_0$ *no depende de* x *si no solo de la tasa de convergencia de* $\sum_{k=1}^{\infty} |a_k| + |b_k|$*, la convergencia de* S_N *es uniforme.*

Polinomio trigonométrico aproximante para funciones de período $\tau \neq 2\pi$

La expresión para el polinomio trigonométrico dada en (4.33) es periódica con período 2π. Para aproximar una función τ-periódica $f(x)$ con $\tau \neq 2\pi$, tendríamos que hacer la siguiente adaptación.

Sea $f(x)$ τ-periódica con $\tau \neq 2\pi$. Definiendo $g(x) = f\left(\frac{\tau x}{2\pi}\right)$, es fácil demostrar que $g(x)$ es una función periódica con período 2π,

$$\begin{aligned} g(x+2\pi) =& f\left(\frac{\tau}{2\pi}(x+2\pi)\right) \\ =& f\left(\frac{\tau x}{2\pi}+\tau\right) \\ =& f\left(\frac{\tau x}{2\pi}\right) \\ =& g(x) \end{aligned} \tag{4.45}$$

Buscamos entonces una función $p(x)$ aproximante para $g(x)$ y se toma $q(x) = p\left(\frac{2\pi}{\tau}x\right)$ como aproximante para $f(x)$, que es de período τ. Es decir

$$f(x) \approx p(\frac{2\pi x}{\tau}) = \frac{a_0}{2} + \sum_{j=1}^{n} a_j cos(\frac{2\pi jx}{\tau}) + b_j sen(\frac{2\pi jx}{\tau}) \tag{4.46}$$

$$= \sum_{j=-n}^{n} c_j e^{ij\frac{2\pi x}{\tau}} \tag{4.47}$$

Ejemplo 63 *Obtener la serie de Fourier para la prolongación periódica de*

$$f(x) = \begin{cases} 1, & si \;\; 0 \leq x \leq \pi \\ -1, & si \;\; \pi < x \leq 2\pi \end{cases}$$

Como esta función es discontinua en $k\pi$, $k \in \mathbb{Z}$, no se puede esperar que sea bien aproximada por bases de funciones que son continuas, además si se tiene en cuenta el teorema de Dirichlet pues la serie de Fourier convergerá a la semisuma de los límites laterales en los puntos de discontinuidad.

Coeficientes exactos de F :

$$\begin{aligned}
c_j^* &= \widehat{f}(j) = \langle f, e^{ijx} \rangle = \frac{1}{2\pi} \int_0^{2\pi} f(x)\, e^{-ijx} dx \\
&= \frac{1}{2\pi} \left[\int_0^{\pi} 1 e^{-ijx} dx + \int_{\pi}^{2\pi} (-1)\, e^{-ijx} dx \right] \\
&= \frac{1}{2\pi} \left[\frac{1}{-ij}\, e^{-ijx}\Big|_0^{\pi} + \frac{1}{ij}\, e^{-ijx}\Big|_{\pi}^{2\pi} \right],\ j \neq 0 \\
&= \frac{1}{2\pi} \left[\frac{1}{-ij} \left(e^{-ij\pi} - e^{0} \right) + \frac{1}{ij} \left(e^{-ij2\pi} - e^{-ij\pi} \right) \right] \\
&= \frac{1}{2\pi} \left[\frac{1}{-ij} \left(\cos \pi j - i sen \pi j - 1 \right) + \frac{1}{ij} \left(\cos 2\pi j - i sen 2\pi j - \cos \pi j + i sen \pi j \right) \right] \\
&= \frac{1}{2\pi} \left[\frac{-i}{i^2 j} \left(\cos \pi j - 1 \right) + \frac{i}{i^2 j} \left(\cos 2\pi j - \cos \pi j \right) \right] \\
&= \frac{1}{2\pi} \left[\frac{i}{j} \cos \pi j - 1 - 1 + \cos \pi j \right] \\
&= \frac{i}{\pi j} \left(\cos \pi j - 1 \right) = \begin{cases} = 0 & j \ par \\ = -\frac{2i}{\pi j} & j \ impar \end{cases}
\end{aligned}$$

Como f es real,

$$\begin{aligned}
c_{-j} &= \overline{c_j} = \overline{\left(-\frac{2i}{\pi j} \right)} = \frac{2}{\pi j} i,\ \textit{para } j \textit{ impar} \\
j &= 0, \quad c_0 = \widehat{f}(0) = 0.
\end{aligned}$$

Por tanto , los coeficientes son

$$\begin{aligned}
c_0 &= \widehat{f}(0) = 0 \\
c_j &= \widehat{f}(j) = \begin{cases} 0, & \text{si } j \text{ par} \\ -\frac{2}{\pi j} i, & \text{si } j \text{ impar} \end{cases} \\
c_{-j} &= \widehat{f}(-j) = \frac{2}{\pi j} i, \qquad \text{si } j \text{ impar.}
\end{aligned}$$

Si expresamos la serie en términos de senos y cosenos se tiene

$$\begin{aligned}
a_j &= c_j + c_{-j} = 0 \\
b_j &= i(c_j - c_{-j}) = \frac{4}{\pi j},
\end{aligned}$$

entonces como sólo aparecen los términos de índice impar

$$f(x) = \frac{4}{\pi} \sum_{j=1}^{\infty} \frac{1}{2j-1} sen(2j-1)x \tag{4.48}$$

Veamos en las figuras (4.4), (4.5), (4.6) y (4.7) cómo a medida que se toman más términos en las sumas parciales N-ésimas pues nos vamos acercando a la forma de la onda cuadrada, sin embargo en las esquinas (puntos de discontinuidad), hay problemas, lo cual es lógico porque como se dijo más arriba se está aproximando una función con discontinuidades mediante una suma finita de funciones continuas. Otro aspecto importante a destacar es el hecho de que para lograr una aproximación cada vez mejor se necesita considerar frecuencias arbitrariamente grandes.

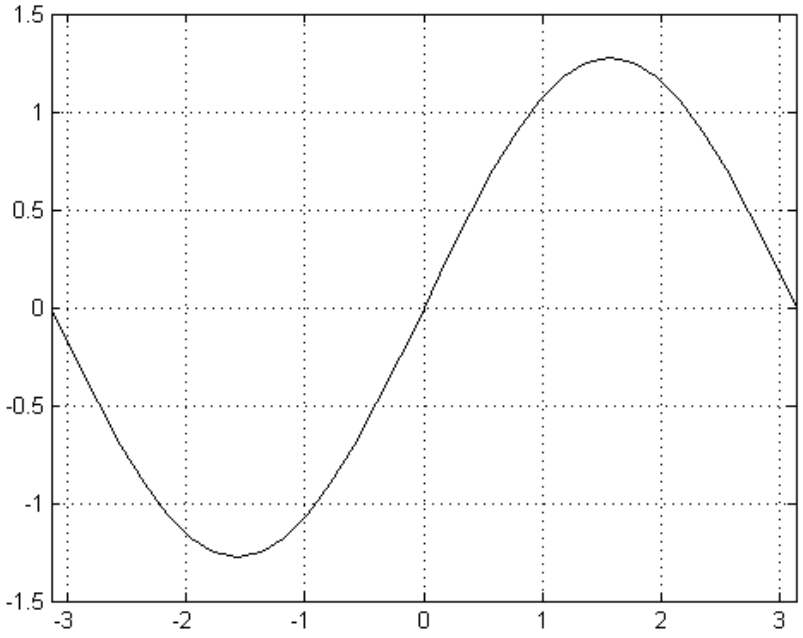


Figura 4.4: $S_1 = \frac{4}{\pi} sen x$

Comentarios

Existen dos características distintivas en la representación de una función f mediante una serie de Fourier:

- la función f se descompone en una suma infinita de componentes mutuamente ortogonales $(g_n(x) := c_n e^{inx})$
- la base ortonormal es generada por la dilatación de una función simple $w(x) := e^{ix}$, $w_n(x) = w(nx) = e^{inx}$ para todo entero n.
- Si interpretamos a la función dada como una señal, entonces los coeficientes de Fourier forman un conjunto denominado espectro de la señal, el espectro nos informa cuánto de cada frecuencia hay en la señal (en el caso del sonido, nos dirá cuanto de cada frecuencia hay en el sonido). Los coeficientes de Fourier para valores pequeños de j nos dan la información de las

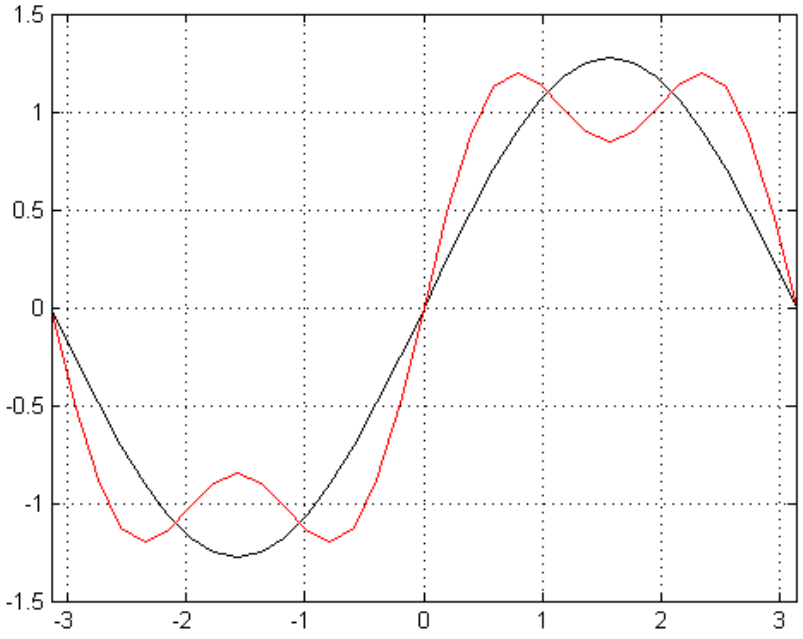


Figura 4.5: $S_1 = \frac{4}{\pi} senx$, $S_2 = \frac{4}{\pi}[senx + \frac{1}{3} sen3x]$

bajas frecuencias y los coeficientes para valores grandes de j nos dan información de las altas frecuencias.

- Estamos ante una función real si y sólo si ella es igual a su conjugada compleja, pero como

$$\overline{\sum_{j=-n}^{n} c_j e^{ijx}} = \sum_{j=-n}^{n} \overline{c_j} e^{-ijx} = \sum_{j=-n}^{n} \overline{c_{-j}} e^{ijx} \tag{4.49}$$

lo que significa que (4.49) es una función real si y sólo si $c_j = \overline{c_{-j}}$ para j. Entonces

$$a_j = 2Rec_j$$
$$b_j = -2Imc_j$$

- El número $\left|\widehat{f}(j)\right|$ mide en cuanto un movimiento armónico simple de frecuencia angular j está presente en el movimiento total. La sucesión completa $\left|\widehat{f}(0)\right|, \left|\widehat{f}(1)\right|, ...$(ó quizás la sucesión de sus cuadrados) es llamada el espectro de potencias o simplemente el espectro de $f(x)$. Note que mediante la relación de Parseval el espectro de $f(x)$ está acotado mediante $\|f\|_2^2$, pero $f(x)$ puede variar ampliamente en su comportamiento en dependencia de como la **energía total** $\|f\|_2^2$ está distribuida sobre el espectro $\left|\widehat{f}(0)\right|, \left|\widehat{f}(1)\right|, ...$ Una función ruidosa tendrá un valor significativo de $\left|\widehat{f}(j)\right|$ para valores mayores de j, mientras que para una función suave el espectro disminuirá rapidamente en la medida que j aumente

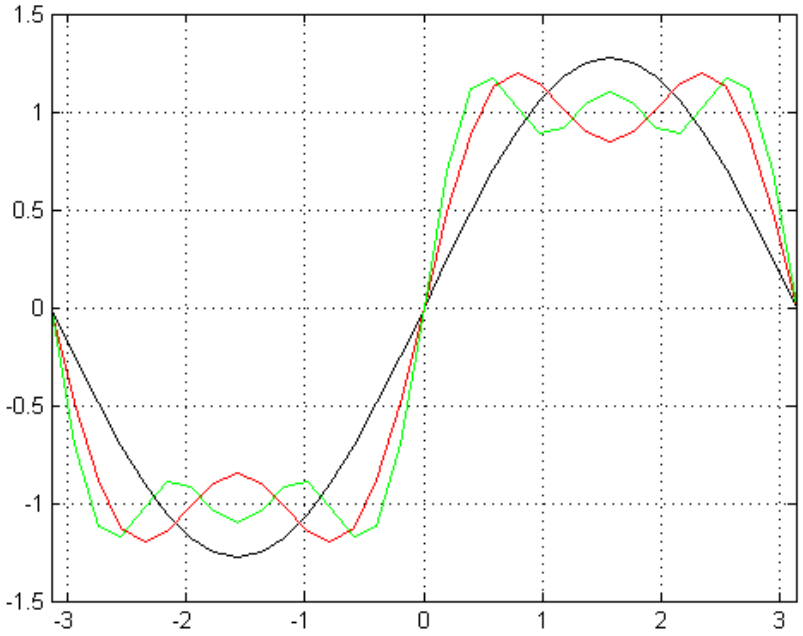


Figura 4.6: $S_1 = \frac{4}{\pi} senx$, $S_2 = \frac{4}{\pi}[senx + \frac{1}{3} sen3x]$, $S_3 = \frac{4}{\pi}[senx + \frac{1}{3} sen3x + \frac{1}{5} sen5x$

- Un método favorito de suavización consiste en generar los coeficientes de Fourier de una función dada $f(x)$ a partir de los datos, filtrando estos coeficientes, lo que significa suprimir ciertas frecuencias, usualmente altas, y entonces reconstruir la función como serie de Fourier con estos coeficientes purificados o filtrados, (ver Fig. 67) para un ejemplo.
- Puede demostrarse que:
  - $$\left|\widehat{f}(j)\right| \to 0$$
  - $$\left|\widehat{f}(j)\right| = O\left(|j|^{-k-1}\right)$$

    si $f(x)$ tiene $k-1$ derivadas continuas y su k-ésima derivada es continua por tramos (o está acotada)

A continuación veremos resultados sobre la convergencia de la serie de Fourier de una función, [página 78 First Course in wavelet with Fourier Analysis].

Teorema 64 *Sea $f \in L^2([-\pi, \pi])$. Entonces para cada x donde la derivada de f está definida, la serie de Fourier de f en x converge a $f(x)$.*

Para demostrar el teorema anterior es suficiente demostrar que la suma parcial n-ésima de la serie de Fourier converge a f cuando $n \to \infty$, es decir

$$\|S_n(x) = p(x) - f\|_{L^2} \longrightarrow 0 \text{ cuando n} \longrightarrow \infty \tag{4.50}$$

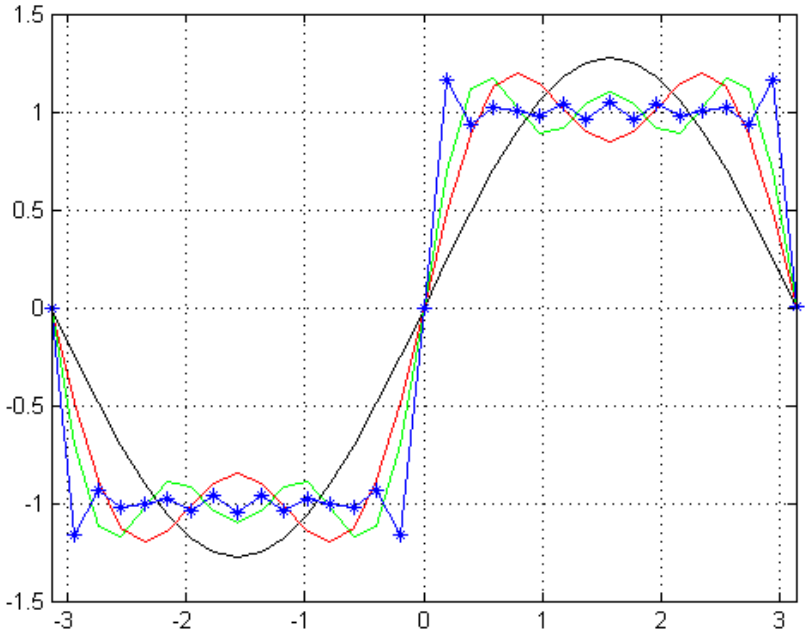


Figura 4.7: S1, S2, S3 y S7

4.4.2. Cálculo aproximado de los coeficientes de Fourier

Usualmente no es posible calcular los coeficientes de Fourier exactamente, debido a que la integral no puede evaluarse en forma cerrada, o porque la función $f(x)$ no se conoce analíticamente. En cualquier caso será necesario aplicar integración numérica. Para nuestros propósitos basta recordar la regla compuesta de los trapecios

$$I(g) = \int_a^b g(x)\,dx \approx T_N = h\sum_{i=1}^{N-1} g_l + \frac{h}{2}(g_0 + g_N) \tag{4.51}$$

$g_l = g(a + lh)$ $\quad h = \frac{b-a}{N} = \frac{2\pi}{N}$, $g(2\pi) = g(0)$ por ser g periódica con período 2π.
Tenemos entonces que para los puntos muestrales $x_l = \frac{2\pi l}{N}$:

$$\int_0^{2\pi} g(x)\,dx \approx \frac{2\pi}{N}\sum_{l=0}^{N-1} g(x_l)$$

Sustituyendo $g(x) = f(x)\,e^{-ijx}$

$$\int_0^{2\pi} f(x)\,e^{-ijx}dx \approx \frac{2\pi}{N}\sum_{l=0}^{N-1} f(x_l)\,e^{-ijx_l}.$$

Denotando la aproximación correspondiente a $\widehat{f}(j)$ como $\widehat{f_N}(j)$, recordar que $\widehat{f}(j) = \frac{1}{2\pi}\int_0^{2\pi} f(x)\,e^{-ijx}dx$,

entonces

$$\widehat{f}(j) \approx \widehat{f_N}(j) = \frac{1}{N}\sum_{l=0}^{N-1} f(x_l)\, e^{-ijx_l} \tag{4.52}$$

donde $x_l = x_0 + lh = \frac{2\pi l}{N}, l = 0, .., N-1$ y la expresión (4.52) coincide con la expresión de c_j para el producto escalar discreto

$$c_j = \langle f, \varphi_j \rangle_N = \widehat{f_N}(j)$$

Estos puntos x_l son llamados puntos de muestra y los números $f(x_l)$ son los correspondientes valores de muestra. Estamos entonces ante la interrogante

¿Cuán bien aproxima $\widehat{f_N}(j)$ a $\widehat{f}(j)$?

Para responder esta pregunta se recuerda primero que las funciones $1, e^{\pm ix}, e^{\pm 2ix}, ...,$ son ortonormales con respecto a el producto escalar discreto,

$$\langle g, h \rangle = \frac{1}{N}\sum_{i=0}^{N-1} g(x_l)\,\overline{h(x_l)}$$

explícitamente

$$\left\langle e^{ikx}, e^{ijx} \right\rangle_N = \begin{cases} 1, & \text{si, } k \equiv j \bmod N \\ 0, & \text{si, } k \neq j \bmod N \end{cases} \tag{4.53}$$

Partiendo entonces de la expresión para calcular $\widehat{f_N}(j)$

$$\widehat{f_N}(j) = \left\langle f, e^{ijx} \right\rangle_N \tag{4.54}$$

y asumiendo que la serie de Fourier $\sum_{j=-\infty}^{\infty} \widehat{f}(j)\, e^{ijx}$ converge absolutamente a $f(x)$ (Como se demostró esto sólo requiere la existencia del límite $\sum_{j=-\infty}^{\infty} \left|\widehat{f}(j)\right|$), y sustituyendo $f(x)$ en (4.54) tenemos que:

$$\begin{aligned} \widehat{f_N}(j) &= \left\langle \sum_{k=-\infty}^{\infty} \widehat{f}(k)\, e^{ikx}, e^{ijx} \right\rangle_N \\ &= \sum_{k=-\infty}^{\infty} \widehat{f}(k) \left\langle e^{ikx}, e^{ijx} \right\rangle_N \end{aligned}$$

Aplicando entonces el resultado (4.53), tendremos que

$$\widehat{f_N}(j) = \sum_{k \equiv j (\bmod N)} \widehat{f}(k) \tag{4.55}$$

Entonces

1. De la expresión (4.55) se puede leer que, teóricamente cada coeficiente aproximado $\widehat{f_N}(j)$ se expresa como la suma de infinitos coeficientes exactos, sin embargo en la práctica para calcular los coeficientes exactos hay que aplicar la regla de los trapecios para aproximar la integral (se obtiene una suma finita para cada coeficiente).

2. La expresión (4.55) no responde a la interrogante: ¿cuántos coeficientes $\widehat{f}_N(j)$ es necesario calcular?.

A continuación veremos dos vías que sustentan una respuesta a la pregunta anterior:

- Los coeficientes $\widehat{f}_N(j)$ son periódicos con período N, es decir comienzan a repetirse una vez que se han calculado los N primeros,

$$\begin{aligned} \widehat{f}_N(j+N) &= \frac{1}{N}\sum_{l=0}^{N-1} f(x_l)\, e^{-i(j+N)x_l} \\ &= \frac{1}{N}\sum_{l=0}^{N-1} f(x_l)\, e^{-i(j\frac{2\pi l}{N})} e^{-i(2\pi l)} \qquad (4.56) \\ &= \widehat{f}_N(j). \end{aligned}$$

  Más aún se cumple $\widehat{f}_N(j+mN) = \widehat{f}_N(j)$ para un entero m y los restantes elementos de la clase de equivalencia no aportan información nueva. Por tanto no tiene sentido considerar todos los elementos de la clase de equivalencia, con un representante basta y se tienen N clases de equivalencia, por tanto se tendrán N coeficientes distintos.

- Por otra parte si $k \equiv j(\mathrm{mod}\, N)$ (otra notación posible es $k \equiv j(N)$), entonces $k = j + mN$ para algún entero m, y para todos los puntos muestrales $x_l = \frac{2\pi}{N} l$ se cumple

$$e^{ikx_l} = e^{i(j+mN)x_l} = e^{ijx_l} e^{imNx_l}$$

  sustituyendo $e^{imN\left(\frac{2\pi}{N}l\right)} = e^{2\pi ml} = \cos(2\pi ml) + isen(2\pi ml) = 1$. Luego

$$e^{ikx} = e^{ijx}, \text{ para } x = x_l = \frac{2\pi l}{N}, \forall l$$

  La igualdad $e^{ikx} = e^{ijx}$ significa que las dos funciones e^{ikx} y e^{ijx} coinciden en todo punto muestral $x = x_l$ usado en el cálculo de $\widehat{f}_N(j)$, si $k \equiv j(\mathrm{mod}\, N)$. Se puede decir que e^{ikx} es un **alias** ó pseudónimo de e^{ijx} en este caso pues son iguales, siendo $k \neq j$, de ahí la terminología aliasing o pseudonimación.

- Es decir para $k = j + mN$, no sólo coinciden los coeficientes $\widehat{f}_N(j) = \widehat{f}_N(k)$, sino que también coinciden las dos funciones e^{ikx} y e^{ijx} en todo punto muestral $x = x_l$. Se reafirma que a lo sumo pueden calcularse N coeficientes $\widehat{f}_N(j)$ distintos (4.55), siendo N el tamaño de la muestra.

- Se obtiene este mismo resultado si empleamos la vía de las ecuaciones normales para calcular los coeficientes del polinomio, ya que si tenemos una muestra de tamaño N, pues sólo tendremos información para plantear N ecuaciones, lo que implica encontrar N coeficientes.

- Si interpretamos a la función f como una señal y al índice j como la frecuencia, entonces el hecho de que $e^{ikx} = e^{ijx}$ para $k = j + mN$, significa que altas frecuencias se pueden confundir con bajas frecuencias en un intervalo, cuando la frecuencia de muestreo no es la adecuada, lo cual es el efecto aliasing.

- Como se demuestra sólo se pueden calcular N coeficientes distintos, repartidos entre valores positivos y negativos de j, entonces para una muestra de N puntos en un intervalo dado $[a, b]$, la frecuencia máxima que se puede representar es $j = \frac{N}{2}$. De ahí que tomando como unidad la longitud del intervalo, la cantidad de puntos muestrales por unidad (frecuencia muestral, $\frac{N}{b-a}$) debe ser mayor que dos veces la frecuencia angular máxima presente en la función a aproximar.

4.4.3. Efecto aliasing. Interpretación práctica

Para evitar el efecto aliasing a la hora de obtener los coeficientes se necesita muestrear $f(t)$ de manera que la frecuencia de muestreo $f_s = \frac{1}{\Delta t}$ para producir $f(n)$ sea al menos mayor que dos veces la frecuencia máxima presente en $f(t)$; veamos un ejemplo.

Sea $f(t) = sen(5 * t)$. Esta función tiene período $T = \frac{2\pi}{5}$, por tanto la frecuencia $f = \frac{5}{2\pi}$ y la frecuencia de muestreo f_s debe cumplir $f_s > 2 * f$.

Si muestreamos la función en $[0, 2\pi]$ siendo N el tamaño de la muestra, entonces la cantidad de muestras que se toma en el intervalo es la frecuencia de muestreo $\frac{N}{2\pi}$ debe ser mayor que dos veces f esto es $\frac{N}{2\pi} > 2 * \frac{5}{2\pi}$ de ahí que $N > 10$.

En la figura (4.8) se grafica el polinomio de orden 2 (para $N = 4$), que como se observa es una representación distorcionada de la función a pesar de que coincide en los puntos muestrales con el polinomio de orden 5 (para $N = 11$)

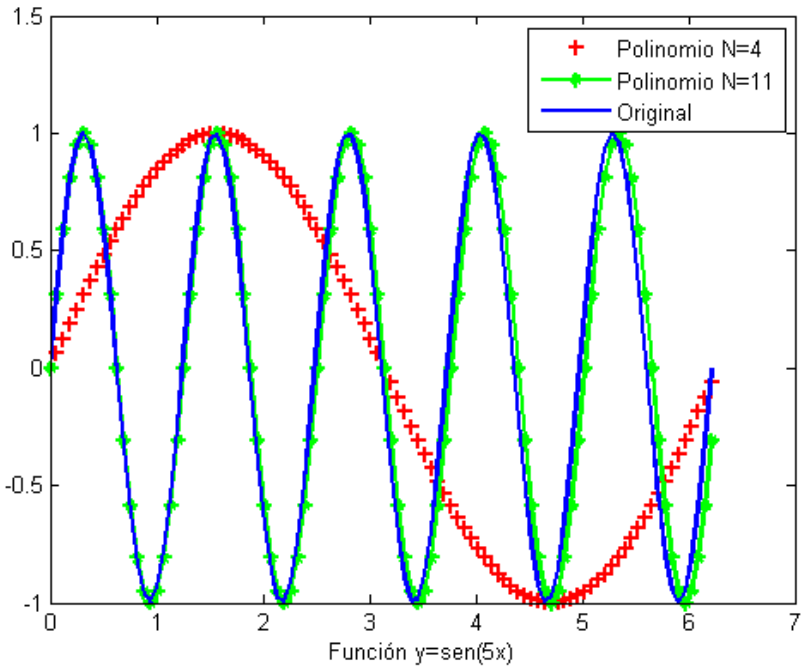


Figura 4.8:

Se puede decir que la frecuencia de resolución Δf es inversamente proporcional al tamaño de la muestra, esto es $\Delta f = \frac{2\pi}{N}$, es decir mientras mayor sea la muestra, se tendrá una mejor resolución

en frecuencias, el proceso de detectar las frecuencias presentes en $f(t)$ es mejor.

Si se muestrea la función en su período principal que en este caso es $\left[0, \frac{2\pi}{5}\right]$ entonces con un razonamiento análogo al anterior se obtiene $N > 2$. Aquí mostramos la gráfica en la figura (4.9) incluyendo $N = 2$ que como se ve está distorcionada

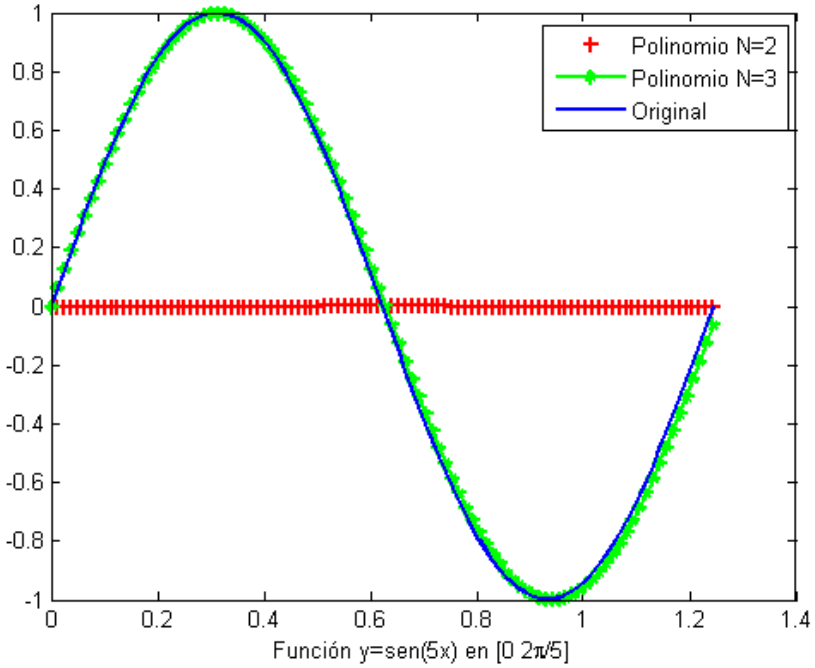


Figura 4.9:

Es importante señalar que cuando se muestrea en N puntos uniformemente espaciados en $[0, 2\pi)$ se acostumbra a identificar la función e^{ijx} con la función $e^{ij'x}$ tal que $j' \equiv j (\mathrm{mod}\, N)$ y cuya frecuencia (angular) $|j'|$ sea la menor posible, quedando j' unívocamente definida por j y N, en otras palabras esto quiere decir que se toma como representante de la clase de equivalencia el de menor valor modular, con la siguiente excepción: cuando N es par y j es un múltiplo impar de $\frac{N}{2}$, entonces tanto $\frac{N}{2}$ como $-\frac{N}{2}$ podrían servir como j', pero en este caso es común escoger, como representante de su clase, el promedio de las dos funciones $e^{i\frac{N}{2}x}$ y $e^{-i\frac{N}{2}x}$, es decir

$$\frac{e^{i\frac{N}{2}x} + e^{-i\frac{N}{2}x}}{2} = \cos\left(\frac{N}{2}x\right), \tag{4.57}$$

lo que quiere decir que tomaríamos $j' = \frac{N}{2}$.

Cuando f es una función suave con $k-1$ derivadas continuas y su k-ésima derivada es continua a trozos, debido al efecto aliasing es inútil calcular $\widehat{f}_N(j)$ como aproximación de $\widehat{f}(j)$ para $|j| > \frac{N}{2}$, con N grande, pues el error es $O\left(N^{-k}\right)$

$$\widehat{f}_N(j) = \widehat{f}(j) + O\left(N^{-k}\right).$$

4.4.4. Expresiones para el polinomio trigonométrico que aproxima a $f(x)$.

Por definición de polinomio trigonométrico teníamos que

$$p(x) = \sum_{j=-n}^{n} c_j e^{ijx} \tag{4.58}$$

En el caso discreto

$$c_j^* = \widehat{f_N}(j) = \left\langle f, e^{ijx} \right\rangle_N = \frac{1}{N} \sum_{l=0}^{N-1} f(x_l)\, e^{-ijx_l}$$

y por otro lado tenemos que según (4.55)

$$\widehat{f}_N(j) = \sum_{k \equiv j (\mathrm{mod}\, N)} \widehat{f}(k)$$

y de (4.56) se tiene que es suficiente con calcular un representante de cada clase de equivalencia, por lo que se calculan N coeficientes distintos en (4.58) y se obtiene que el polinomio trigonométrico que aproxima a $f(x)$ es:

$$p(x) = \sum_{|j|<\frac{N}{2}} \widehat{f}_N(j)\, e^{ijx}, \text{ para } N \text{ impar}, N = 2n+1 \tag{4.59}$$

$$p(x) = \sum_{|j|<\frac{N}{2}} \widehat{f}_N(j)\, e^{ijx} + \widehat{f}_N\left(\frac{N}{2}\right) \mathrm{Re}\left(e^{i\frac{N}{2}x}\right), \text{ para } N \text{ par}, N = 2n \tag{4.60}$$

Los sumandos de $p(x)$ reciben el nombre de armónicos. El orden n del polinomio aproximante puede tomarse entonces entre 0 y $\left[\frac{N}{2}\right]$.

Observación 65 *: El segundo término en (4.60) aparece ya que cuando N es par y j es un múltiplo impar de $\frac{N}{2}$, la función representativa de la clase de equivalencia se toma según (4.57)*

Ejemplo 66 *Sea $N = 4$.*

x_0	x_1	x_2	x_3	
0	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$	2π
$f(x_0)$	$f(x_1)$	$f(x_2)$	$f(x_3)$	

El efecto aliasing tiene como consecuencia que en la sumatoria que define $\widehat{f}_N(j)$ basta considerar las clases de congruencia módulo 4, que son las siguientes:

$$\{..., -12, -8, -4, 0, 4, 8, 12, ...\} \to e^0 = 1 \to j = 0$$
$$\{..., -11, -7, -3, 1, 5, 9, 13, ...\} \to e^{ix} \to j = 1$$
$$\{..., -9, -5, -1, 3, 7, 11, 15, ...\} \to e^{-ix} \to j = -1$$

y en esta última $\{...,-10,-6,-2,2,6,10,14,...\}$ *tendríamos que* j *es múltiplo impar de* $\frac{N}{2}$ *y como* N *es par, siguiendo (4.57) sumaríamos* j *dede* -1 *hasta* 1 *y le agregaríamos el término* $\widehat{f}_N\left(\frac{N}{2}\right)\cos\left(\frac{N}{2}x\right)$. *Para este ejemplo se tomarían:* $j=0,1,2,-1$ *quedando unívocamente definida por* j *y* N *la función* e^{ijx} *representativa de la clase. Según el efecto aliasing teóricamente cada coeficiente* $\widehat{f}_N(j)$ *está compuesto por la suma de infinitos coeficientes exactos* $\widehat{f}(k)$ *de Fourier :*

$\widehat{f}_4(j)=...+\widehat{f}(j-8)+\widehat{f}(j-4)+\widehat{f}(j)+\widehat{f}(j+4)+\widehat{f}(j+8)+...$

donde $j=0,1,2,-1$

$$\begin{aligned}\widehat{f}_4(0)&=...+\widehat{f}(-8)+\widehat{f}(-4)+\widehat{f}(0)+\widehat{f}(4)+\widehat{f}(8)+...\\ \widehat{f}_4(1)&=...+\widehat{f}(-7)+\widehat{f}(-3)+\widehat{f}(1)+\widehat{f}(5)+\widehat{f}(9)+...\\ \widehat{f}_4(2)&=...+\widehat{f}(-6)+\widehat{f}(-2)+\widehat{f}(2)+\widehat{f}(6)+\widehat{f}(10)+...\\ \widehat{f}_4(-1)&=...+\widehat{f}(-5)+\widehat{f}(-1)+\widehat{f}(3)+\widehat{f}(7)+\widehat{f}(11)+...\end{aligned}$$

aunque ya vimos que en la práctica hay que usar integración aproximada para calcularlos (regla de los trapecios)

$$\widehat{f}_4(j)=\frac{1}{4}\sum_{l=0}^{3}\left(f(x_l)\,e^{-ijx_l}\right),j=0,1,2,-1,\ \text{o sea}\ |j|\leq\frac{N}{2}$$

entonces

$$p(x)=\widehat{f}_4(-1)\,e^{-ix}+\widehat{f}_4(0)\,e^{0}+\widehat{f}_4(1)\,e^{ix}+\widehat{f}_4(2)\cos(2x)$$

El efecto **aliasing** no permite observar frecuencias mayores que $N/2$ para una muestra de tamaño N.

Observación 67 *En el caso (4.60) de* N *par, si* $f(x)$ *es además real puede probarse debido al efecto* ***aliasing****, que* $p(x)$ *interpola a* $f(x)$ *en los puntos muestrales* $\left\{x_l=\frac{2\pi l}{N}\right\}$, *es decir* $p(x_l)=f(x_l),l=0,..,N-1$

En el caso de N impar, $N=2n+1$ se tiene el siguiente resultado

Teorema 68 *Para cualquier* $m\leq n$ *el polinomio trigonométrico de orden* m

$$p(x)=\sum_{j=-m}^{m}\widehat{f}_N(j)\,e^{ijx}$$

es la mejor aproximación a $f(x)$ *mediante polinomios trigonométricos de orden* m *con respecto a la norma de mínimos cuadrados discreta*

$$\|g\|_2=\left(\langle g,g\rangle_N\right)^{\frac{1}{2}}=\left(\frac{1}{N}\sum_{l=0}^{N-1}\left|g\left(\frac{2\pi l}{N}\right)\right|^2\right)^{\frac{1}{2}}$$

En particular para $m=n$ *, el polinomio trigonométrico de orden* n

$$p(x)=\sum_{j=-n}^{n}\widehat{f}_N(j)\,e^{ijx}$$

interpola a $f(x)$ *en los puntos muestrales* $x_l=\frac{2\pi l}{N}$, *para toda* l.

Demostración 69

$$p(x) = \sum_{j=-n}^{n} \widehat{f}_N(j)\, e^{ijx}$$

$$p(x) = \sum_{j=-n}^{n} \frac{1}{N} \sum_{k=0}^{N-1} f(x_k) e^{-ijx_k} e^{ijx}$$

Entonces

$$p(x_l) = \sum_{j=-n}^{n} \frac{1}{N} \underbrace{\sum_{k=0}^{N-1} f(x_k) e^{-ijx_k} e^{ijx_l}}_{f(x_l),\ para\ k=l}$$

$$\begin{aligned} p(x_l) &= \sum_{j=-n}^{n} \frac{1}{N} f(x_l) \\ &= \frac{(2n+1)}{N} f(x_l) \\ &= f(x_l),\ para\ N\ impar \end{aligned}$$

Cuando $f(x)$ es una función real, puede escribirse el polinomio de interpolación en forma real.

Teorema 70 *(de Riemann-Lebesgue) Sea f una función seccionalmente continua en el intervalo $[a,b]$. Entonces*

$$\lim_{k\to\infty} \int_a^b f(x)\cos(kx)dx = \lim_{k\to\infty} \int_a^b f(x) sen(kx)dx = 0$$

Dejamos la demostración al lector, pero lo más importante es que este resultado precisamente es la base de la compresión, ya que solo un número finito de los coeficientes de la serie de Fourier serán mayores que un determinado valor prefijado.

4.5. Funciones periódicas con período $2T$

Teorema 71 *El conjunto de funciones $\left\{ \frac{1}{\sqrt{2T}} e^{\frac{in\pi x}{T}}, n = ... -2, -1, 0, 1, 2, ... \right\}$ es una base ortonormal para $L_2[-T,T]$. Si*

$$f(x) = \sum_{j=-\infty}^{\infty} c_j e^{\frac{i\pi jx}{T}}$$

entonces $c_j = \frac{1}{2T} \int_{-T}^{T} f(x) e^{-\frac{i\pi jx}{T}} dx$

4.6. Transformada discreta de Fourier

En secciones anteriores se estudió la serie trigonométrica y el polinomio trigonométrico de Fourier desde dos puntos de vista: i) como una forma de aproximar funciones periódicas usando bases ortogonales, ii)como una herramienta matemática para analizar el contenido de frecuencia de una **señal periódica**. Se discutió cuántos coeficientes era necesario calcular al construir el polinomio trigonométrico aproximante. Para el cálculo de los coeficientes se utilizaron expresiones que pueden ser interpretadas como la Transformada Integral o Continua de Fourier para el caso continuo y como la Transformada Discreta de Fourier para el caso discreto. Sin embargo en la práctica generalmente estos coeficientes se calculan de forma aproximada usando valores discretos, por lo que se discutirá en más detalle el algoritmo de la Transformada Discreta de Fourier y la forma eficiente de implementarlo.

Definición 72 *Para cada entero positivo N y para cada arreglo $\overrightarrow{f} \in C^N$, la transformada discreta de Fourier de $\overrightarrow{f}$ es el arreglo $\widehat{f}$ también de longitud N*

$$\widehat{f}_j = \langle f, \overrightarrow{\omega_j} \rangle_N = \frac{1}{N} \sum_{l=0}^{N-1} f_l \left(\omega_N^j\right)^l, j = 0, 2, ..N-1 \tag{4.61}$$

donde $\omega_N := e^{\frac{-2\pi i}{N}}$.

Hay textos que se refieren a (4.61) como la transformada finita de Fourier. En lo que sigue veremos con más detalle quién es ω_N, pero primero señalemos dos aspectos importantes: i) los datos de entrada, vector de componentes f_l deben ser tomados en puntos equidistantes ii) la longitud del vector de datos se considera como el período de la función, por lo que para trabajar en el intervalo $[0, 2\pi]$ es necesario realizar un escalamiento.

Es fácil ver que la evaluación de cualquier $\widehat{f_N}(j)$ requiere N sumas y N multiplicaciones, esto si consideramos $e^{-ijx_l} = q_l$ previamente calculado. El cálculo directo de estos N números $\widehat{f_N}(j)$ requerirían $2N^2$ operaciones en punto flotante.

Por ejemplo si se desea obtener los coeficientes de Fourier de $f(x)$ con un orden de precisión de $\epsilon = 10^{-6}$, conociendo que $f(x) \in C^1(\mathbb{R})$ y tiene segunda derivada continua a trozos. Cuántos nodos muestrales se necesitan?. Por el teorema visto anteriormente $\left|\widehat{f_N}(j) - \widehat{f}(j)\right| = O(N^{-2})$. Luego $N^{-2} \approx 10^{-6}$, con lo que $N \approx 10^3$, es decir se necesitarían 1000 puntos muestrales y requeriríamos un millón de operaciones lo cual constituía un obstáculo mayor para el uso del análisis de Fourier discreto.

Varias personas llegaron de forma independiente a obtener algoritmos rápidos de la transformada discreta de Fourier, conocidos como la transformada rápida de Fourier (TRF), pero se considera como punto de partida la propuesta por Cooley y Tukey en 1965. Se ha podido comprobar que existen muchas TRF que dependen de la descomposición de N. Los algoritmos modernos de la transformada rápida de Fourier tienen complejidad computacional $O(Nlog_2N)$, en lugar de $O(N^2)$

Hacer el cálculo de los coeficientes $\widehat{f_N}(j)$ es lo que se denomina hacer un análisis armónico discreto de Fourier, donde interesa, dada f, qué valores de j (índice de frecuencias) están presentes así como su magnitud. Para el cómputo de dichos coeficientes, a partir de los números $f(x_0), ..., f(x_{N-1})$, se usa la transformada discreta de Fourier F_N.

La transformada discreta de Fourier se puede interpretar como una transformación lineal de los valores de la función f, esto es $\widehat{f_N} = \frac{1}{N} F \cdot f$. En otras palabras, la transformada discreta de Fourier expresa arreglos de datos reales ó complejos como combinación lineal de otros arreglos ordenados por frecuencias, en lugar de los valores iniciales de los datos (es decir la TDF significa cambiar de coordenadas en el espacio).
Como $e^{-2\pi i} = 1$ entonces $(e^{\frac{-2\pi i}{N}})^N = 1$, lo que implica que las potencias $e^{\frac{-2\pi i}{N}}$ se denominan raíces $N-$ complejas de la unidad. Entonces denotando $\omega_N := e^{\frac{-2\pi i}{N}}$ y $\overrightarrow{\omega_j} := (e^{-ijx_l})_{l=0}^{N-1}$, donde la coordenada con índice l en el arreglo $\overrightarrow{\omega_j}$ es igual a la raíz N-ésima de la unidad $\omega_N := e^{-i2\pi/N}$. Consideremos $(\overrightarrow{\omega_j})_l = \left(e^{-i2\pi/N}\right)^{jl} = \omega_N^{jl}$.

Por tanto si tenemos como conjunto de datos una sucesión finita de números complejos $f_0, ..., f_{N-1}$, que se corresponde con el vector $\overrightarrow{f} = (f_0, ..., f_{N-1}) \in C^N$, con respecto a la base canónica:

$$\overrightarrow{e}_0 = (1, 0, ..., 0), \overrightarrow{e}_1 = (0, 1, ..., 0), .. \overrightarrow{e}_{N-1} = (0, 0, ..., 1_{N-1})$$

La transformada discreta de Fourier expresa un arreglo $\overrightarrow{f}$ no con sus valores f_l, si no como una combinación lineal de arreglos del tipo $\overrightarrow{\omega_j} := (e^{-ijx_l})_{l=0}^{N-1}$.

Ejemplo 73 *Si $N = 1$, entonces $\omega_N = \omega_1 = e^{-i2\pi} = 1$ y $\overrightarrow{\omega_0} = (\omega_1^0) = 1$, en este caso $\widehat{f_0} = f_0$*

Ejemplo 74 *Si $N = 2$, entonces $\omega_N = \omega_2 = e^{-i2\pi/2} = -1$*

$$\begin{aligned} \overrightarrow{\omega_0} &= \left(\left[\omega_2^0\right]^0, \left[\omega_2^0\right]^1\right) = (1, 1) \\ \overrightarrow{\omega_1} &= \left(\left[\omega_2^1\right]^0, \left[\omega_2^1\right]^1\right) = (1, -1) \end{aligned}$$

$$F = \begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix} \tag{4.62}$$

Entonces $\overrightarrow{\widehat{f}} = \frac{1}{N} F \cdot \overrightarrow{f}$, esto es $\widehat{f_0} = \frac{f_0 + f_1}{2}$ y $\widehat{f_1} = \frac{f_0 - f_1}{2}$. Para $N = 2$, la matriz de Fourier coincide con la matriz de Haar (se verá más adelante).

Como se observa para calcular la transformada discreta de Fourier cada producto matriz vector cuesta N^2 multiplicaciones. Es precisamente el hecho de que los arreglos $\overrightarrow{\omega_j}$ sean mutuamente ortogonales con respecto a una variante del producto escalar usual, lo que nos facilita un cálculo rápido de la trasformada discreta de Fourier. Veamos el siguiente ejemplo

Ejemplo 75 *Si $N = 4$, entonces $\omega_N = \omega_4 = e^{-i\pi/2} = -i$, $j = 0, 1, 2, 3$ y para cada j, $l = 0, 1, 2, 3$*

$$\begin{aligned} \overrightarrow{\omega_0} &= \left(\left[\omega_4^0\right]^0, \left[\omega_4^0\right]^1, \left[\omega_4^0\right]^2, \left[\omega_4^0\right]^3\right) = (1, 1, 1, 1) \\ \overrightarrow{\omega_1} &= \left(\left[\omega_4^1\right]^0, \left[\omega_4^1\right]^1, \left[\omega_4^1\right]^2, \left[\omega_4^1\right]^3\right) = (1, -i, -1, i) \\ \overrightarrow{\omega_2} &= \left(\left[\omega_4^2\right]^0, \left[\omega_4^2\right]^1, \left[\omega_4^2\right]^2, \left[\omega_4^2\right]^3\right) = (1, -1, 1, -1) \\ \overrightarrow{\omega_3} &= \left(\left[\omega_4^3\right]^0, \left[\omega_4^3\right]^1, \left[\omega_4^3\right]^2, \left[\omega_4^3\right]^3\right) = (1, i, -1, -i) \end{aligned}$$

En este último ejemplo la matriz F de la transformación lineal sería

$$F = \begin{matrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{matrix} \tag{4.63}$$

Se observa que los N^2 coeficientes de la matriz F solo pueden tomar N valores distintos, pero además como $e^{-2\pi i} = 1$ entonces

$$\begin{aligned} (\omega_N)^N &:= \left(e^{\frac{-2\pi i}{N}}\right)^N = 1, \\ (\omega_N)^{N+1} &= \omega_N^1 \end{aligned}$$

y en general

$$\omega_N^k = \omega_N^j, \text{ si } k \equiv j mod N \tag{4.64}$$

es decir $k = m \cdot N + j, m \in \mathbb{Z}$.
La propiedad anterior repercute en lo siguiente: para $N = 4$ por ejemplo, en las componentes de índice par $(j = 2k)$ de la transformada discreta, el coeficiente de f_0 es igual al de f_2 y el coeficiente de f_1 es igual al de f_3

$$\begin{aligned} \widehat{f_0} &= \frac{1}{4}(f_0 + f_1 + f_2 + f_3) \\ \widehat{f_2} &= \frac{1}{4}(f_0 - f_1 + f_2 - f_3) \end{aligned}$$

y podemos escribir

$$\begin{aligned} \widehat{f_{2k}} &= \sum_{l=0}^{\frac{N}{2}-1} \omega_N^{2kl} \left(f_l + f_{\frac{N}{2}+l}\right) \\ &= \sum_{l=0}^{\frac{N}{2}-1} \left(\omega_N^2\right)^{kl} \left(f_l + f_{\frac{N}{2}+l}\right), \quad k = 0, 1 \\ &= \sum_{l=0}^{\frac{N}{2}-1} \left(\omega_N^2\right)^{kl} y_l \end{aligned} \tag{4.65}$$

Es decir los coeficientes de índice par $\widehat{f_{2k}}$, $k = 0, .., \frac{N}{2} - 1$ se obtienen al aplicar la transformada discreta de Fourier al vector $y_l := f_l + f_{\frac{N}{2}+l}$, que es un arreglo de dimensión $\frac{N}{2}$ (la mitad de la dimensión del arreglo original). Para las componentes de índice impar de $\widehat{f}$

$$\begin{aligned} \widehat{f_1} &= \frac{1}{4}(f_0 - if_1 - f_2 + if_3) \\ \widehat{f_3} &= \frac{1}{4}(f_0 + if_1 - f_2 - if_3) \end{aligned}$$

y si recordamos que $\omega_N = \omega_4 = -i$, entonces podemos escribir

$$\begin{aligned} \widehat{f}_{2k+1} &= \sum_{l=0}^{\frac{N}{2}-1} \omega_N^{2kl} \omega_N^l \left(f_l - f_{\frac{N}{2}+l} \right) \\ &= \sum_{l=0}^{\frac{N}{2}-1} \left(\omega_N^2 \right)^{kl} y_{\frac{N}{2}+l}, \quad k = 0, .., \frac{N}{2} - 1 \end{aligned} \tag{4.66}$$

donde $y_{\frac{N}{2}+l} := \omega_N^l \left(f_l - f_{\frac{N}{2}+l} \right)$, $l = 0, .., \frac{N}{2} - 1$. Es decir los coeficientes de índice impar se obtienen de aplicar la transformada discreta de Fourier al vector $y_{\frac{N}{2}+l} := \omega_N^l \left(f_l - f_{\frac{N}{2}+l} \right)$ de dimensión $\frac{N}{2}$.

Para comprender mejor la idea, veamos que el nuevo vector $\overrightarrow{y}$ que se construye tiene la misma cantidad de componentes que el vector original, lo que sucede es que con la primera mitad de las componentes de $\overrightarrow{y}$ se calculan las transformadas de las componentes de índice par del vector original y con la segunda mitad de las componentes de $\overrightarrow{y}$ se calculan las transformadas de las componentes de índice impar. En el ejemplo que estamos analizando el vector de entrada tiene 4 componentes, entonces con las componentes y_0 y y_1 del nuevo vector se calculan las transformadas de las componentes de índice par del vector original, $\widehat{f_0}$, $\widehat{f_2}$ y con y_2 y y_3 se calculan las transformadas de las componentes de índice impar del vector original, $\widehat{f_1}$, $\widehat{f_3}$.

A partir de este momento con todas las subdivisiones posteriores que se hagan, solo se alterará el orden de las transformadas porque ya en la primera mitad quedaron todos los índices pares y en la segunda mitad todos los impares. Este efecto no se observa en el ejemplo que estamos tratando ya que el próximo paso sería dividir cada vector de dos componentes en dos vectores con una componente cada uno. Más abajo se verá un ejemplo con más datos iniciales.

Por lo tanto encontrar de forma eficiente (disminuyendo el costo computacional) la transformada discreta de Fourier $\overrightarrow{\widehat{f}} = F \cdot \overrightarrow{f}$ de un vector $\overrightarrow{f}$ de dimensión N , $\widehat{f_j} = \frac{1}{N} \sum_{l=1}^{N-1} f_l \left(\omega_N^j \right)^l, j = 0, .., N-1$ se transforma en encontrar dos transformadas discretas (4.65) y (4.66) de dimensión $\frac{N}{2}$ más $O(\frac{N}{2})$ operaciones (las N sumas y $\frac{N}{2}$ multiplicaciones) para calcular los nuevos vectores $y_l := f_l + f_{\frac{N}{2}+l}$ y $y_{\frac{N}{2}+l}$). A su vez estas se pueden reducir a dos transformaciones donde la dimensión se reduce de nuevo a la mitad y así sucesivamente. En esto se basa la transformada rápida de Fourier.

Si denotamos por $C(N)$ la complejidad computacional para realizar la TDF de un vector de datos de longitud $N = 2^P$, en la forma explicada más arriba, entonces en el primer paso

$$C(N) = 2C(\frac{N}{2}) + 3\frac{N}{2}$$

en el segundo paso

$$C(\frac{N}{2}) = 2C(\frac{N}{4}) + \frac{3N}{4}$$

con lo cual

$$\begin{aligned} C(N) &= 2(2C(\frac{N}{4}) + \frac{3N}{4}) + \frac{3N}{2} \\ &= 4C(\frac{N}{4}) + \frac{3N}{2} + \frac{3N}{2} \\ &\vdots \\ &= P \cdot \frac{3N}{2} + N \\ &= \frac{3N}{2} log_2 N + N \end{aligned}$$

Ejemplo 76 *: Dada el vector de datos para $N = 8$,*

0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$3\frac{\pi}{4}$	π	$5\frac{\pi}{4}$	$3\frac{\pi}{2}$	$7\frac{\pi}{4}$
5	1	2	8	2	5	8	1

Calcule la Transformada Discreta de Fourier

En este caso $\omega_N = \omega_8 = e^{-i\pi/4} = \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}$, $j = 0, 1, 2, 3, 4, 5, 6, 7$ y para cada j, $l = 0, 1, 2, 3, 4, 5, 6, 7$.

Para simplificar la notación digamos $a = \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}$.
Entonces tenemos que calcular primeramente los nuevos vectores

$$y_l = f_l + f_{\frac{N}{2}+l} \tag{4.67}$$

$$y_{l+\frac{N}{2}} = a^l(f_l - f_{\frac{N}{2}+l}), \tag{4.68}$$

con $l = 0, .., \frac{N}{2} - 1$

$$\begin{aligned} y_0 &= 5 + 2 = 7 \\ y_1 &= 1 + 5 = 6 \\ y_2 &= 2 + 8 = 10 \\ y_3 &= 8 + 1 = 9 \end{aligned}$$

$$\begin{aligned} y_4 &= (5 - 2)a^0 = 3 \\ y_5 &= (1 - 5)a^1 = -4a \\ y_6 &= (2 - 8)a^2 = 6i \\ y_7 &= (8 - 1)a^3 = 7(-\overline{a}) \end{aligned}$$

Como se observa el nuevo vector $\overrightarrow{y}$ tiene 8 componentes igual que el vector de datos original, lo que sucede es que con las primeras componentes y_0, y_1, y_2, y_3 se calculan las componentes de índice par de la transformada de f, $\widehat{f}_0, \widehat{f}_2, \widehat{f}_4, \widehat{f}_6$ y con la otra mitad de las componentes de $\overrightarrow{y}$ se calculan las componentes de índice impar de la transformada de f, $\widehat{f}_1, \widehat{f}_3, \widehat{f}_5, \widehat{f}_7$.

Ahora recursivamente a partir de cada vector de dimensión cuatro anterior obtenemos dos vectores de dimensión dos, es decir en las fórmulas anteriores ahora $N = 4$ y $l = 0, 1$ y por tanto $\omega_N = \omega_4 = e^{-i\pi/2} = -i$, que no es más que a^2. Entonces del primer vector de dimensión 4 se obtiene un vector de dimensión 4 también

$$\begin{aligned} z_0 &= y_0 + y_2 = 7 + 10 = 17 \\ z_1 &= y_1 + y_3 = 6 + 9 = 15 \end{aligned}$$

$$\begin{aligned} z_2 &= (7 - 10)(a^2)^0 = -3 \\ z_3 &= (6 - 9)(a^2)^1 = 3i \end{aligned}$$

pero ahora con las dos primeras componentes se calculan las componentes de índice par de la transformada del vector de dimensión 4, $\widehat{f_0}, \widehat{f_2}, \widehat{f_4}, \widehat{f_6}$ que serán $\widehat{f_0}, \widehat{f_4}$, nos fijamos que ya aquí se produce una alteración en la ordenación. Con la otra mitad del vector $\overrightarrow{z}$ se calculan las componentes de índice impar de la transformada del vector $\widehat{f_0}, \widehat{f_2}, \widehat{f_4}, \widehat{f_6}$, que serán $\widehat{f_2}, \widehat{f_6}$, es decir nos seguimos moviendo dentro de las componentes de índice par de la trasnformada del vector original y reiteramos, solo va cambiando el orden. Ahora con la nueva subdivisión de la segunda parte pues seguimos calculando las componentes de índice impar del vector original, con z_4, z_5

$$\begin{aligned} z_4 &= y_4 + y_6 = 2 + 6i \\ z_5 &= y_5 + y_7 = -4a - 7\overline{a} \end{aligned}$$

se calcula la transformada de las componentes de índice par del vector $\widehat{f_1}, \widehat{f_3}, \widehat{f_5}, \widehat{f_7}$ que serán $\widehat{f_1}, \widehat{f_5}$ (cambio de ordenación).

Con la segunda mitad, es decir con z_6, z_7

$$\begin{aligned} z_6 &= (y_4 - y_6)(a^2)^0 = 2 - 6i \\ z_7 &= (y_5 - y_7)(a^2)^1 = (-4a + 7\overline{a})(-i) \end{aligned}$$

se calcula la transformada de las componentes de índice impar del vector $\widehat{f_1}, \widehat{f_3}, \widehat{f_5}, \widehat{f_7}$ que serán $\widehat{f_3}, \widehat{f_7}$.

Finalmente de cada vector de dimensión dos anterior, obtenemos dos de dimensión uno, que serán las 8 componentes de la transformada de Fourier del vector inicial. Aquí $\omega_N = \omega_2 = e^{-i\pi} = -1$, que no es más que a^4, entonces

$$\begin{aligned} \widehat{f_0} &= z_0 + z_1 = 15 + 17 = 32 \\ \widehat{f_4} &= (z_0 - z_1)(a^4)^0 = (15 - 17)(a^4)^0 = 2 \end{aligned}$$

$$\begin{aligned} \widehat{f_2} &= z_2 + z_3 = -3 + 3i \\ \widehat{f_6} &= (z_2 - z_3)(a^4)^0 = -3 - 3i \end{aligned}$$

$$\begin{aligned}\widehat{f_1} &= z_4 + z_5 = (3 - 11\frac{\sqrt{2}}{2}) + i(6 - 3\frac{\sqrt{2}}{2}) \\ \widehat{f_5} &= (z_4 - z_5)(a^4)^0 = (3 + 11\frac{\sqrt{2}}{2}) + i(6 + 3\frac{\sqrt{2}}{2})\end{aligned}$$

$$\begin{aligned}\widehat{f_3} &= z_6 + z_7 = (\frac{6 + 11\sqrt{2}}{2}) + i(\frac{-12 - 3\sqrt{2}}{2}) \\ \widehat{f_7} &= (z_6 - z_7)(a^4)^0 = (\frac{6 - 11\sqrt{2}}{2}) + i(\frac{-12 + 3\sqrt{2}}{2})\end{aligned}$$

Se deja de tarea al lector realizar la transformación por vía directa.

4.7. Transformada rápida de Fourier.

La propiedad $e^{w+z} = e^w e^z$ para todos los números w y z conduce a una reducción del número de operaciones aritméticas necesarias para calcular la transformada discreta de Fourier, mediante un algoritmo publicado en 1965 por James W. Cooley y John W. Turket, conocida como la transformada rápida de Fourier (FFT)

La expresión (4.61) no es más que la evaluación de un polinomio algebraico de grado N en la variable w_N^j con coeficientes f_l. Aplicando el algoritmo de Horner al cálculo con variable compleja esta evaluación se lleva a cabo con N^2 operaciones (N polinomios de grado N), entendiéndose por operación una multiplicación compleja con una suma compleja.

En el algoritmo desarrollado por Cooley y Tukey, la evaluación de los coeficientes aproximados de Fourier consume sólo $O(N \log N)$ operaciones, de ahí su denominación. En muchas áreas de aplicación (telecomunicaciones, análisis de series de tiempo, acústica, música electroacústica, etc.) este algoritmo ha ocasionado un cambio de actitud respecto al uso de la aproximación trigonométrica en las computadoras ya que en lugar de N^2 operaciones bastan $N(r_1 + r_2 + .. + r_p)$ si $N = r_1 r_2 .. r_p$.

En general si N es par el cálculo de la transformada discreta se reduce al cálculo de dos transformadas de longitud $\frac{N}{2}$, si además de ser par es potencia de dos, $N = 2^p$, entonces el proceso se puede repetir y la transformada rápida de dimensión N se calcula en término de dos transformadas de dimensión $\frac{N}{2}$, éstas a su vez se transforman en cuatro de dimensión $\frac{N}{4}$, éstas en ocho de dimensión $\frac{N}{8}$ y así hasta alcanzar N de dimensión 1. Una transformada rápida de dimensión uno es el propio valor. El número de pasos en la recursión en este caso es p. El número de operaciones en cada paso es una $O(N)$ por tanto el costo total es una $O(Np) = O(N \log_2 N)$. Si N no es potencia de 2 de todas formas es posible expresar la transformada rápida de Fourier en varias de menor diemensión. Si N no es un número primo entonces de la original de dimensión N se pueden obtener dos de dimensión que divida a N. Si N es un número primo entonces se puede incluir en otra cuya dimensión pueda ser factorizada. Podemos concluir diciendo que la transformada rápida de Fourier emplea la estrategia divide y vencerás.

4.7.1. Transformada rápida de Fourier hacia adelante.

El siguiente resultado nos da una transformación recursiva de la propiedad de la exponencial mencionada anteriormente ($e^{w+z} = e^w e^z$)

Definición: Para cada entero positivo par N, para cada arreglo $\overrightarrow{f} \in \mathbb{C}^N$, definamos los correspondientes arreglos ${}_{par}\overrightarrow{f}$ e ${}_{impar}\overrightarrow{f}$ que tienen la mitad de los elementos de $\overrightarrow{f}$: para cada índice $m \in \left\{0, .., \left(\frac{N}{2} - 1\right)\right\}$

$$\begin{aligned}
{}_{par}x_m &: = x_{2m} = 2m.\frac{2\pi}{N} = m2\pi/\left(N/2\right) \\
{}_{impar}x_m &: = x_{2m+1} = \left(2m+1\right).\frac{2\pi}{N} = \frac{m2\pi}{N/2} + \frac{2\pi}{N} \\
\left({}_{par}\overrightarrow{f}\right)_m &: = \overrightarrow{f}_{2m} = f\left(2m\frac{2\pi}{N}\right) \\
\left({}_{impar}\overrightarrow{f}\right)_m &: = \overrightarrow{f}_{2m+1} = f\left(\left[2m+1\right],2\pi/N\right)
\end{aligned}$$

Si denotamos la transformada discreta de Fourier (DFT) de $\overrightarrow{f}$, como $\widehat{f}$, entonces

$$\begin{aligned}
{}_{par}\widehat{f} &= \text{DFT de } {}_{par}\overrightarrow{f} \\
{}_{impar}\widehat{f} &= \text{DFT de } {}_{impar}\overrightarrow{f}
\end{aligned}$$

Lema: Para cada $k \in \left\{0, .., \left(\frac{N}{2} - 1\right)\right\}$,

$$\begin{aligned}
\widehat{f}_k &= \frac{1}{2}\left({}_{par}\widehat{f}_k + \left[e^{-i2\pi/N}\right]^k \left[{}_{impar}\widehat{f}_k\right]\right) \\
\widehat{f}_{k+(N/2)} &= \frac{1}{2}\left({}_{par}\widehat{f}_k - \left[e^{-i2\pi/N}\right]^k \left[{}_{impar}\widehat{f}_k\right]\right)
\end{aligned}$$

Demostración: Consideremos dos casos. En cada caso dividamos la suma que define la Transformada discreta de Fourier en una suma con índices pares y en una suma con índices impares. En el primer caso para cada índice $k \in \left\{0, .., \left(\frac{N}{2} - 1\right)\right\}$

$$\begin{aligned}
\widehat{f}_k &= \left\langle \overrightarrow{f}, \overrightarrow{\omega}_k \right\rangle_N = \frac{1}{N}\sum_{m=0}^{N-1} f_m e^{-ikm2\pi/N} \\
&= \frac{1}{N}\sum_{m=0}^{(N/2)-1} f_{2m} e^{-ik[2m]2\pi/N} + \frac{1}{N}\sum_{m=0}^{(N/2)-1} f_{2m+1} e^{-ik[2m+1]2\pi/N} \\
&= \frac{1}{2}\frac{1}{N/2}\sum_{m=0}^{(N/2)-1} f_{2m} e^{-ik[2m]\pi/N/2} + \frac{1}{2}\frac{1}{N/2}\sum_{m=0}^{(N/2)-1} f_{2m+1} e^{-ik[2m]\pi/N/2}.e^{-ik\pi/N/2} \\
&= \frac{1}{2}\left({}_{par}\widehat{f}_k + \left[e^{-i2\pi/N}\right]^k \left[{}_{impar}\widehat{f}_k\right]\right)
\end{aligned}$$

De forma similar, para cada índice $k \in \{0, .., (\frac{N}{2} - 1)\}$

$$\widehat{f}_{k+(N/2)} = \left\langle \widehat{f}, \overrightarrow{\omega}_{k+\frac{N}{2}} \right\rangle_N$$
$$= \frac{1}{N} \sum_{m=0}^{N-1} f_m e^{-i\left[k+\frac{N}{2}\right]m2\pi/N}$$

$$= \frac{1}{N} \sum_{m=0}^{(N/2)-1} f_{2m} e^{-i[k+N/2][2m]2\pi/N}$$
$$+ \frac{1}{N} \sum_{m=0}^{(N/2)-1} f_{2m+1} e^{-i[k+N/2][2m+1]2\pi/N}$$

$$= \frac{1}{2}\frac{1}{N/2} \sum_{m=0}^{(N/2)-1} f_{2m} e^{-i[k+N/2][2m]\pi/N/2}$$
$$+ \frac{1}{2}\frac{1}{N/2} \sum_{m=0}^{(N/2)-1} f_{2m+1} e^{-i[k+N/2][2m]\pi/N/2} . e^{-i[k+N/2]\pi/N/2}$$
$$= \frac{1}{2}\left({}_{par}\widehat{f}_k - \left[e^{-i2\pi/N}\right]^k \left[{}_{impar}\widehat{f}_k \right] \right)$$

Ejemplos:

1. Para $N = 1 \quad \overrightarrow{f} = f_0$, $f_0 \in \mathbb{C}$ →existe un único $\overrightarrow{\omega_0} = 1$, por tanto la Transformada Rápida de Fourier transforma $\overrightarrow{f}$ en $\widehat{f}$ sin cambio

$$\widehat{f}_0 = \left\langle \overrightarrow{f}, \overrightarrow{\omega_0} \right\rangle_1 = \frac{1}{1} f_0 \overline{1} = f_0$$

2. Para $N = 2$ cada arreglo $\overrightarrow{f} = (f_0, f_1)$ se divide en dos arreglos ${}_{par}\overrightarrow{f} = f_0$ e ${}_{impar}\overrightarrow{f} = f_1 \Rightarrow$

$$\begin{aligned} {}_{par}\widehat{f}_0 &= {}_{par}f_0 = f_{2,0} = f_0 \\ {}_{impar}\widehat{f}_0 &= {}_{impar}f_0 = f_{2,0+1} = f_1 \end{aligned}$$

Entonces con $e^{-i0\pi/(N/2)} = 1$, aplicando recursión

$$\begin{aligned} \widehat{f}_0 &= \frac{1}{2}\left({}_{par}\widehat{f}_0 + 1 \left[{}_{impar}\widehat{f}_0 \right] \right) \\ &= \frac{f_0 + f_1}{2} \end{aligned}$$

$$\begin{aligned} \widehat{f}_{1+0} &= \frac{1}{2}\left({}_{par}\widehat{f}_0 - 1 \left[{}_{impar}\widehat{f}_0 \right] \right) \\ &= \frac{f_0 - f_1}{2} \end{aligned}$$

obteniéndose los mismos resultados que para la DFT.

3. Para $N = 4$ cada arreglo $\overrightarrow{f} = (f_0, f_1, f_2, f_3) := (5, 1, 2, 8)$. Un primer paso de recursión divide el arreglo inicial en dos arreglos

$$\begin{aligned} {}_{par}\overrightarrow{f} &= (f_0, f_2) = (5, 2) \\ {}_{impar}\overrightarrow{f} &= (f_1, f_3) = (1, 8) \end{aligned}$$

del ejemplo anterior, teníamos que para ${}_{par}\overrightarrow{f}$

$$\begin{aligned} {}_{par}\widehat{f}_0 &= \frac{1}{2}\left({}_{par}f_0 + {}_{par}f_1\right) \\ &= \frac{f_0 + f_2}{2} \\ &= \frac{7}{2} \end{aligned}$$

$$\begin{aligned} {}_{par}\widehat{f}_1 &= \frac{1}{2}\left({}_{par}f_0 - {}_{par}f_1\right) \\ &= \frac{f_0 - f_2}{2} \\ &= \frac{3}{2} \end{aligned}$$

Entonces

$$\begin{aligned} {}_{par}\overrightarrow{f} &= (5, 2) \\ &= \frac{7}{2}(1, 1) + \frac{3}{2}(1, -1) \end{aligned}$$

Para ${}_{impar}\overrightarrow{f}$

$$\begin{aligned} {}_{impar}\widehat{f}_0 &= \frac{1}{2}\left({}_{impar}f_0 + {}_{impar}f_1\right) \\ &= \frac{f_1 + f_3}{2} \\ &= \frac{9}{2} \end{aligned}$$

$$\begin{aligned} {}_{impar}\widehat{f}_1 &= \frac{1}{2}\left({}_{impar}f_0 - {}_{impar}f_1\right) \\ &= \frac{f_1 - f_3}{2} \\ &= -\frac{7}{2} \end{aligned}$$

$$\begin{aligned} {}_{impar}\overrightarrow{f} &= (1, 8) \\ &= \frac{9}{2}(1, 1) + \left(-\frac{7}{2}\right)(1, -1) \end{aligned}$$

Entonces con $N = 4$ y $N/2 = 2$, la recursión se aplica a los dos arreglos ${}_{par}\widehat{f} = \left(\frac{7}{2}, \frac{3}{2}\right)$ y ${}_{impar}\widehat{f} = \left(\frac{9}{2}, -\frac{7}{2}\right)$

$$\begin{aligned}
\widehat{f}_0 &= \frac{1}{2}\left({}_{par}\widehat{f}_0 + {}_{impar}\,\widehat{f}_0\right) \\
&= \frac{1}{2}\left(\frac{7}{2} + \frac{9}{2}\right) \\
&= \frac{11}{4} \\
\widehat{f}_2 &= \frac{1}{2}\left({}_{par}\widehat{f}_0 - {}_{impar}\,\widehat{f}_0\right) \\
&= \frac{1}{2}\left(\frac{7}{2} - \frac{9}{2}\right)
\end{aligned}$$

$$\begin{aligned}
\widehat{f}_1 &= \frac{1}{2}\left({}_{par}\widehat{f}_1 - i_{impar}\widehat{f}_1\right) \\
&= \frac{1}{2}\left(\frac{3}{2} + i\frac{7}{2}\right) \\
\widehat{f}_3 &= \frac{1}{2}\left({}_{par}\widehat{f}_1 + i_{impar}\widehat{f}_1\right) \\
&= \frac{1}{2}\left(\frac{3}{2} - i\frac{7}{2}\right)
\end{aligned}$$

4.7.2. Transformada rápida de Fourier con MatLab

En MatLab existe el comando **fft** que calcula la transformada discreta de Fourier eficientemente, es decir no es más que un algoritmo de la transformada rápida de Fourier (**f**ast **f**ourier **t**ransform); aunque al principio aparece sólo una **f,** según uno de los fundadores del MatLab, Cleve Moler esta **f** está tanto por fast como por finite . Para usar este comando solo es necesario pasarle como argumento el vector de las muestras de la función correspondiente $\overrightarrow{f} = (f_0, ..., f_{N-1}) \in C^N$, dichos valores son muestreados en un intervalo de tiempo $t = (0 : N-1) * T_s/N$, donde T_s es el tiempo final. Muchas veces se toma como paso para la discretización temporal el parámetro conocido como tasa de muestreo F_s (muestras por unidad de tiempo); $F_s = \frac{1}{\Delta t}$. Para otras especificaciones ver en la ayuda de MatLab.

En el caso de que la **fft** sea usada para determinar los coeficientes del polinomio trigonométrico aproximante de una función dada, entonces el arreglo de valores que se pasa se supone dado en el intervalo del período principal de la función $[0, T]$ y los puntos muestrales se toman de la forma $t = (0 : N-1) * (T/N)$, siendo T el período y N el tamaño de la muestra, pero el algoritmo de Matlab transforma a período 2π. Se observa que no se tiene en cuenta el valor de la función en el último punto, debido a que ya se supone que es igual al primero.

También se tiene el comando **interpft**, que precisamente calcula el polinomio trigonométrico para los valores muestrales dados también en un intervalo de periodicidad de la función, este comando en sí da la evaluación del polinomio en los puntos que se le pasen **interpft(Y,100)** por ejemplo evalua el polinomio trigonométrico que aproxima a Y en 100 puntos. Están además los comandos

ifft Calcula la transformada inversa discreta.

fftn Calcula la transformada discreta de dimensión n.

ifftn Calcula la transformada inversa discreta de dimensión n.

fftw Permite optimizar el cálculo de la transformada discreta.

fftshift Reorganiza una transformada de modo tal que la frecuencia 0 quede en el medio del arreglo.

4.7.3. Funciones con diferentes frecuencias

Cuando la frecuencia de una función (señal) cambia con el tiempo, ver figura (4.10) se dice

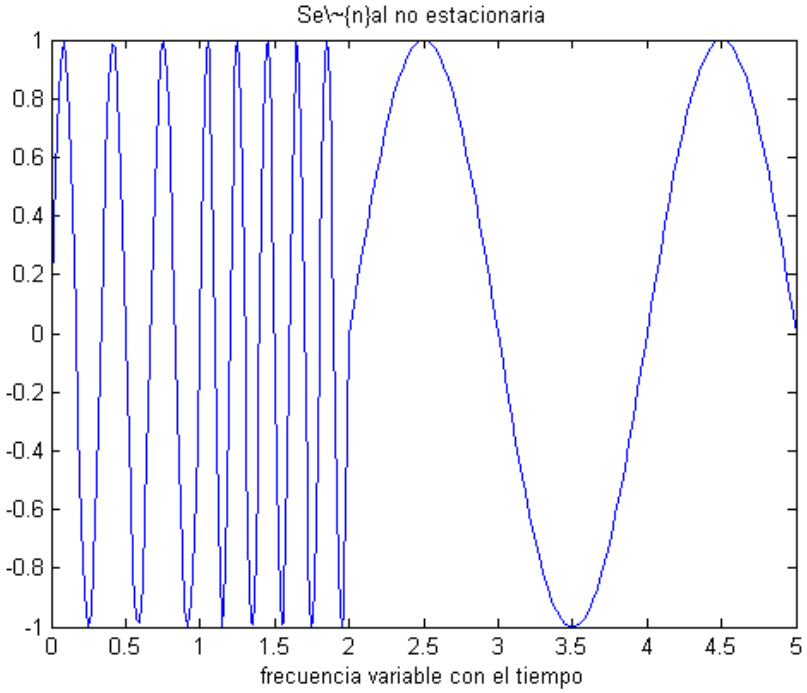


Figura 4.10:

que estamos en presencia de una función no estacionaria. En este caso la serie de Fourier o la transformada de Fourier nos informa sobre las frecuencias presentes en la función, pero no se sabe en qué momento se produce este cambio de frecuencia, es decir hay localización en frecuencia pero no en el tiempo.

Con el objetivo de obtener también una localización temporal para cada frecuencia surge en la década de los 80 una nueva forma de aproximar funciones(transformar funciones) conocida como aproximación mediante bases wavelet (transformada wavelet), lo cual se aborda en el próximo capítulo.

4.8. Comandos en MatLab

Para una implementación computacional en MatLab vea los comandos **fft**, **ifft**, **interpft** entre otros

4.9. Ejercicios para el estudio independiente

1. Dada la función f por la siguiente tabla

x	998	999	1000	1001	1002
f(x)	3.7	4.2	5.1	5.9	6.2

   Encuentre la función de aproximación mínimo cuadrática

   a) en el conjunto conjunto $\varphi = \{1, x\}$,

   b) Para el conjunto $\varphi = \{p_o(x), p_1(x)\} = \{1, x - \overline{x}\} = \{1, x - 1000\}$ formado por funciones ortogonales.

2. Dada la función $y = sen((\pi * x)/5) + x/5$ en los 23 puntos $x = -5 : 0{,}5 : 6$. Determine los polinomios de ajuste mínimo cuadrático aumentando el orden hasta 20. Grafique los de grado 1,3, 6 y 19. Comente sobre la calidad de las aproximaciones y fundamente.

3. Cálculo de una órbita planetaria.

   Dada la forma cuadrática

$$z = ax^2 + bxy + cy^2 + dx + ey + f$$

   el conjunto de puntos (x, y) donde $z = 0$ representa una sección cónica, que puede ser una elipse, una parábola o una hipérbola. Círculos y líneas son casos especiales. La ecuación $z = 0$ se puede normalizar dividiendo por cualquier coeficiente distinto de cero.

   Supongamos que tenemos 10 observaciones de las posiciones que ocupa la órbita que sigue un planeta en el plano (x,y)

x	1.02	.95	.87	.77	.67	.56	.44	.30	.16	.01
y	0.39	.32	.27	.22	.18	.15	.13	.12	.13	.15

   a) Determine los coeficientes de la forma cuadrática que ajusta estos datos en el sentido de mínimos cuadrados, haciendo uno de los coeficientes igual a 1 y resolviendo el sistema sobredeterminado de 10×5. Grafique la órbita tomando las x sobre el eje x y las y sobre el eje y. En el mismo gráfico muestre los 10 puntos de los datos.

   b) El problema de mínimos cuadrados anterior es casi de rango deficiente, para ver el efecto que esto causa en la solución perturbe los datos ligeramente añadiendo a cada coordenada de cada punto en los datos un valor aleatorio con distribución uniforme en el intervalo [-.005, .005]. Calcule los nuevos coeficientes y plotee la órbita sobre el mismo gráfico de la órbita anterior. Comente la comparación.

4. Supongamos que como parte del proceso de admisión en una Universidad se desea predecir el promedio académico p de los estudiantes de nuevo ingreso, basados en:

   - promedio en el preuniversitario x_1,
   - prueba de aptitud en oralidad x_2,
   - prueba de aptitud general cuantitativa x_3

   Basados en datos de estudiantes ya admitidos, se puede construir un modelo lineal

   $$p = \sum_{i=1}^{3} c_i x_i$$

   Encuentre las expresiones de c_i por aproximación lineal múltiple.

5. Modelo de demanda anual per cápita de carne de cerdo en determinado país:

   $$C_t = c_0 + c_1 * P_t + c_2 * I_t,$$

   donde

   - C_t es el consumo per cápita de carne de cerdo en el año t (en kg. por persona)
   - P_t es el precio de la carne en el año t (en pesos por kg.)
   - I_t es el ingreso per cápita en el año t (en miles de pesos)

6. A continuación se muestra la demanda de energía eléctrica en la ciudad de Puerto Montt, desde el año 2004 hasta el 2010, medida en mega watts. El Jefe de Operaciones de la empresa, debe pronosticar la demanda para el 2011 ajustando una recta de tendencia a estos datos:

Año	Demanda de energía eléctrica
2004	74
2005	79
2006	80
2007	90
2008	105
2009	142
2010	122

   La recta de tendencia está dada por: $y = a + bx$, donde a: ordenada, b: pendiente de la recta de regresión.

   Profesionales de Estadística han desarrollado ecuaciones que se utilizan para encontrar los valores de a y b para cualquier recta de regresión. Estas son:

   $$b = \frac{\sum_{i=0}^{N} x_i y_i - n\overline{x}\overline{y}}{\sum_{i=0}^{N} x_i^2 - n\overline{x}^2}$$

   $$a = \overline{y} - b\overline{x}$$

   $\overline{x}$: promedio del valor de las x, $\overline{y}$: promedio del valor de las y, n:cantidad de observaciones

a) Deduzca las fórmulas utilizadas por el Jefe de Operaciones para determinar los valores de a y de b.

b) Halle la recta de regresión correspondiente a estos datos.

c) ¿Cuál debió ser la demanda de energía eléctrica correspondiente al año 2011? ¿Cuál será la correspondiente al 2012?.

d) Grafique la demanda histórica y la recta de tendencia.

Capítulo 5

Introducción a la aproximación con funciones wavelet

En el tema anterior estudiamos algunos aspectos del análisis de Fourier a los que llegamos a través de la aproximación mínimo cuadrática con funciones base ortogonales. Específicamente buscando funciones base ortogonales para aproximar funciones periódicas y de cuadrado integrables en $[-\pi, \pi]$ es decir funciones periódicas en $L^2[-\pi, \pi]$. Se estudió que la herramienta matemática estándar para representar señales periódicas es la serie trigonométrica de Fourier. En los casos que esto no es posible se usan los polinomios trigonométricos aproximantes, sin embargo aún cuando la serie de Fourier converge a la función (señal) que se quiere aproximar, se toma una cantidad finita de términos para la representación, obteniéndose una mejor aproximación a medida que se consideran más términos. La representación en serie de Fourier de una señal periódica nos permite conocer cuáles son las frecuencias j que están presentes en la misma, y los coeficientes $\widehat{f}(j)$ nos dan la amplitud de dichas frecuencias,

$$f(x) = \sum_{j=-\infty}^{\infty} \widehat{f}(j) e^{ijx}.$$

Para el caso de funciones no periódicas y continuas $f : \mathbb{R} \to \mathbb{C}$, la transformada continua de Fourier $\widehat{f} : \mathbb{R} \to \mathbb{C}$

$$\widehat{f}(j) = \int_{-\infty}^{\infty} f(t) e^{-ijt} dt \tag{5.1}$$

nos brinda esta información (contenido de frecuencias), y cuando lo que se tiene es una muestra discreta y finita de la señal entonces la herramienta a usar es la transformada discreta de Fourier,

$$\widehat{f}_N(j) = \sum_{l=0}^{N-1} f_l \left(\omega_N^j\right)^l, j = 0, 2, ..N-1.$$

calculada mediante su versión más eficiente, el algoritmo de la transformada rápida de Fourier y donde $\omega_N = e^{-\frac{2\pi i}{N}}$. En cualquiera de los casos anteriores para obtener la amplitud de una frecuencia en particular hay que conocer la función en todo el eje real, ya que la exponencial compleja está definida en todo el eje real. De aquí que no sea posible saber en qué intervalo de tiempo se produce

cada frecuencia, es decir se tiene muy buena localización en frecuencias pero ninguna localización en tiempo. Esta información es importante principalmente cuando se tienen señales no estacionarias es decir cuando las frecuencias presentes cambian con el tiempo, ver figura (5.1). Observar que si tenemos otra señal con el mismo contenido de frecuencia pero que ocurren en distintos tiempos, ver figura (5.2), pues obtenemos que los coeficientes distintos de cero que se calculan con la transformada rápida de Fourier corresponden a las mismas frecuencias, según muestra la figura (5.3). Entonces el análisis de Fourier es ideal para estudiar datos estacionarios (datos cuyas propiedades estadísticas son invariantes en el tiempo); pero no es bueno para estudiar datos con eventos transitorios; que no pueden predecirse estadísticamente a partir de datos del pasado. Con tales datos no estacionarios en mente fueron diseñadas las que hoy conocemos como funciones wavelets, pensando que si la función se podía descomponer en ondas que no fueran puramente ondas sinusoidales, se podría condensar la información en el dominio de tiempo y en el dominio de frecuencias.

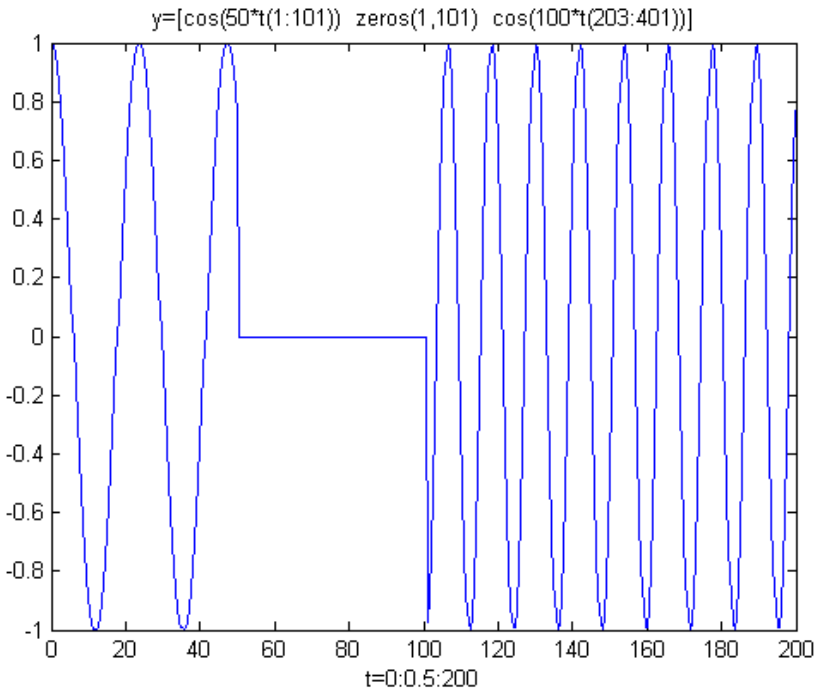


Figura 5.1:

Por otro lado desde el punto de vista más matemático se necesitaba buscar un sistema de funciones base ortogonal para aproximar las funciones en $L^2(\Re) = \{f : \mathbf{R} \longrightarrow \mathbf{C}, \int_{-\infty}^{\infty} |f(x)|^2 dx < \infty\}$. Es decir se necesitaban funciones de corta duración y de ahí nacieron las wavelets. En un lenguaje coloquial diremos que las wavelets son funciones con un comportamiento oscilatorio y con duración limitada es decir con soporte compacto y valor medio igual a cero es decir $\int_{\mathbb{R}} \Psi(x)dx = 0$ y por tanto $\widehat{\Psi}(0) = \int_{\mathbb{R}} \Psi(x)dx = 0$.

Aunque el origen de las funciones que hoy llamamos wavelet se puede remontar a la polémica que desató el postulado presentado por Fourier [1] en 1807 a la Real Academia de Paris- " Una

[1] Jean-Baptiste Joseph Fourier, matemático y físico francés (Auxerre, 1768 – Paris, 1830)

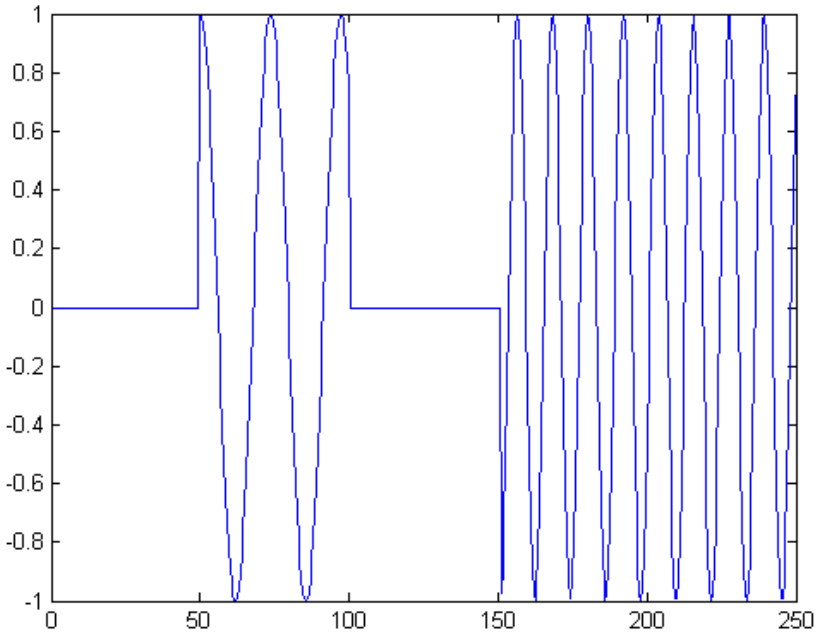


Figura 5.2:

función arbitraria continua o con discontinuidades definida en un intervalo finito por un gráfico arbitrariamente caprichoso siempre puede ser expresada como suma de sinusoides" - y que transformó el mundo; [2] la primera función conocida que cumple con los requisitos de lo que hoy conocemos como función wavelet (término acuñado en 1984 por el ingeniero francés Jean Morlet) se debe al matemático húngaro Alfred Haar (1885-1933), quien en 1910 en su tesis de doctorado introdujo las funciones que hoy se conocen como wavelets de Haar. Su motivación fue encontrar una base para $L^2[0,1]$ que a diferencia del sistema trigonométrico tuviera convergencia uniforme de las sumas parciales para funciones continuas en $[0,1]$ (para las bases de Fourier lo mejor que se puede esperar para funciones continuas es la convergencia casi donde quiera). Después de Haar se incluyeron en la lista nombres como el de Dennis Gabor (físico inglés-húngaro) que en 1946 introdujo la transformada de Gabor, que es la función que tiene la mayor localización simultánea posible en tiempo frecuencia. En los 70's y los 80's la comunidad de procesamiento de imágenes y señales introdujo su propia versión de análisis wavelet pasando por términos como subband coding, quadratur mirror filters y algoritmo piramidal. En 1984 la teoría wavelet adquirió su propia identidad. Jean Morlet tratando de ayudar a los geólogos a encontrar vías más eficaces para encontrar petróleo, desarrolló una forma propia de analizar las señales sísmicas para crear componentes que estuvieran bien localizadas en el espacio y llamó a estas componentes onditas de forma constante (más tarde conocidas como las wavelets de Morlet). De hecho la primera vez que apareció acuñado el término wavelet fue en una artículo de Morlet y el físico francés Alex Grossmann publicado en 1984 [3]. Yves Meyer también uno

[2]Fourier, J.-B. J. *Théorie analytique de la chaleur.* Firmin Didot, Pére et Fils, Paris, 1822.

[3]Grossmann, A., Morlet, J. (1984) Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM Journal of Mathematical Analysis* **15**, 723-736.

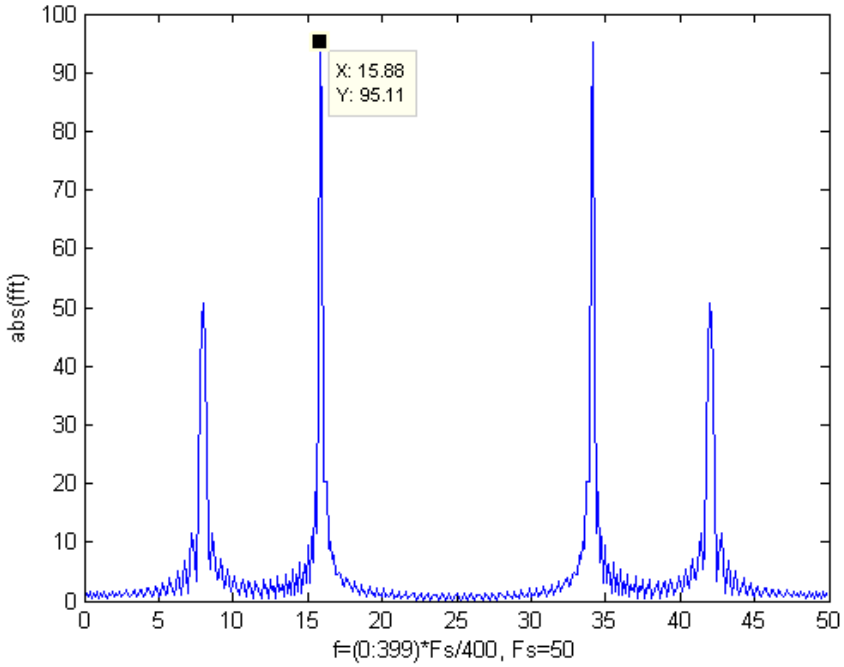


Figura 5.3:

de los precursores de la teoría wavelet, fue el primero en establecer o en darse cuenta de la conexión entre la wavelet de Morlet y otras wavelets anteriores a ésta, además introdujo por primera vez las funciones wavelets ortogonales. En 1986 Stéphane Mallat, estudiante de doctorado de Meyer, en Visión Computacional, relacionó la teoría de wavelets con la literatura existente en subband coding y quadrature mirror filters, que es la versión de wavelets en el procesamiento de imágenes. Mallat y Meyer demostraron que las wavelets aparecen implícitas en el análisis multirresolución. Un último detonante importante de mencionar en la revolución wavelet ocurrió en 1987 cuando Ingrid Daubechies descubrió una nueva clase de wavelet que no era solo ortogonal si no que podía ser implementada usando ideas de filtrado digital simple. Las wavelets de Daubechies no se expresan mediante funciones elementales si no que se construyen de manera recursiva.

La transformada wavelet continua (TWC) es un tipo de transformada que nos brinda una representación tiempo (espacio)-frecuencia (escala) de la función,

$$(W_\Psi f)(a,b) := |a|^{-1/2} \int_{-\infty}^{\infty} f(x)\overline{\Psi(\frac{x-b}{a})}dx$$

donde $W_\Psi : L^2(\mathbb{R}) \to L^\infty(\mathbb{R}_{\not\epsilon} \times \mathbb{R})$ [4] es una aplicación lineal, b posición en el tiempo o en el espacio y a representa la escala o frecuencia. La palabra wavelet significa onda pequeña u ondita pero matemáticamente la interpretación está referida a que la duración de la función es muy limitada, es decir que está muy bien localizada, vea la diferencia en la figura (5.4).

[4] L^∞, espacio de las funciones medibles acotadas

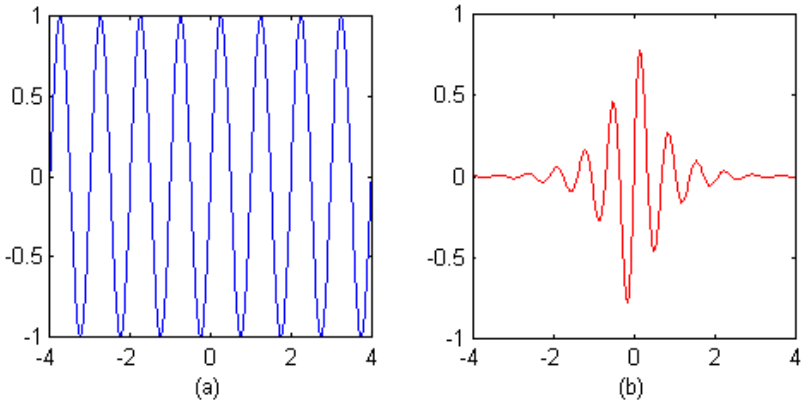


Figura 5.4: a) onda, b) ondita(wavelet), oscilación amortiguada

De forma análoga al análisis de Fourier aquí se puede realizar la síntesis de la función mediante la trasnformada inversa

$$f(x) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_\Psi f)(a,b)\psi(\frac{x-b}{a})\frac{dadb}{2}$$

con $C_\psi = \int_{-\infty}^{\infty} \frac{\left|\widehat{\psi}(j)\right|^2}{j} dj < \infty$, lo cual es posible solo si $\widehat{\Psi}(0) = \int_{\mathbb{R}} \Psi(x)dx = 0$.

La contribución fundamental de la teoría de las wavelets ha sido unificar ideas similares provenientes de diversos campos de aplicación y crear un equilibrio entre dichas técnicas cuyo resultado ha sido una poderosa herramienta de técnicas algorítmicas con una fuerte base teórica. En el análisis numérico el desarrollo de wavelets está relacionado con la aproximación con funciones spline; para una información más detallada consultar [Eugenio Hernández, Guido Weiss; A First Course on Wavelets, CRC Press 1996.]

La teoría de las funciones wavelets como se aprecia es muy versátil y muy rica en conceptos matemáticos y con variadas aplicaciones. Para un estudio conciente de la TW se requiere de conocimientos matemáticos teóricos que se van fuera del currículo de la carrera Ciencia de la Computación. Nosotros nos centraremos en aspectos básicos de la transformada wavelet discreta (TWD), tomando como base la wavelet de Haar.

5.1. Transformada wavelet discreta de Haar

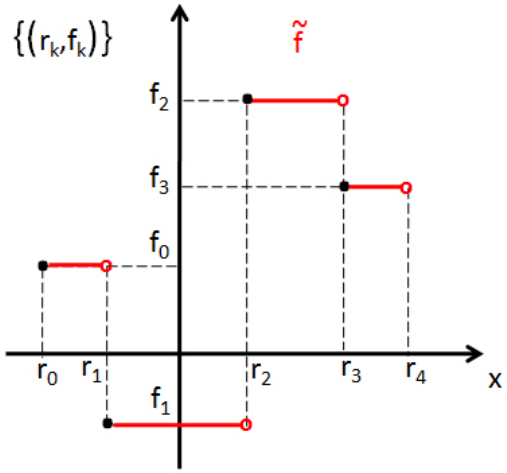
Para analizar y sintetizar una señal o fenómeno físico, (que puede estar dado por un conjunto de datos), aplicando wavelet, el primer paso consiste en representar la señal por una función matemática. En el caso en que los datos provienen de mediciones experimentales, se cuenta en general con una sucesión finita de valores llamada muestra, por tanto de lo que se trata es de representar dicho conjunto de datos por una muestra única. La forma más simple de realizar dicha aproximación es mediante una función escalonada $\widetilde{f}$. Por ejemplo si consideramos los datos $f = (f_0, f_1, f_2, f_3)$, entonces

$$\widetilde{f} = f_0\varphi_{[r_0,r_1[} + f_1\varphi_{[r_1,r_2[} + f_2\varphi_{[r_2,r_3[} + f_3\varphi_{[r_3,r_4[}$$

donde

$$\varphi_{[u,w[}(x) := \begin{cases} 1 & \text{si } u \leq x < w \\ 0 & \text{si no} \end{cases}$$

De esta manera se cuenta con una representación de f en todo el intervalo $[r_0, r_4]$ como se muestra en la figura (5.1) y no sólo en puntos discretos, Considerando que la muestra tiene una cantidad de



elementos que es una potencia entera de dos, se representa en el intervalo $[0, 1]$, y se tiene,

$$\widetilde{f} = f_0\varphi_{[0,1/4[} + f_1\varphi_{[1/4,1/2[} + f_2\varphi_{[1/2,3/4[} + f_3\varphi_{[3/4,1[}.$$

Como se observa la función $\widetilde{f}$ se representa como combinación lineal de traslaciones enteras $k, k \in \{0, .,2^J - 1\}$, de una dilatación diádica de la función característica

$$\varphi_{[0,1[}(x) := \begin{cases} 1 & \text{si } 0 \leq x < 1 \\ 0 & \text{si no} \end{cases}$$

$\varphi_{[0,\frac{1}{4}[}(x) = \varphi_{[0,1[}(2^2x)$, $\varphi_{[\frac{1}{4},\frac{1}{2}[}(x) = \varphi_{[0,1[}(2^2x - 1)$, $\varphi_{[\frac{1}{2},\frac{3}{4}[}(x) = \varphi_{[0,1[}(2^2x - 2)$. Entonces para caracterizar el promedio y variabilidad de los datos, a Haar [5] se le ocurrió expresar la función aproximada $\widetilde{f}$ reemplazando un par de escalones adyacentes por un sólo escalón más amplio, adicionándole lo

[5] Alfred Haar (1885-1933), matemático húngaro

que más tarde se llamó una wavelet. El escalón más amplio mide el promedio del par de escalones iniciales, mientras que la wavelet está formada por dos escalones alternados y mide la diferencia entre el par de escalones iniciales. Por ejemplo la suma de dos escalones adyacentes de ancho $\frac{1}{2}$ produce la función escalón unitaria básica $\varphi_{[0,1)}$,

$$\varphi_{[0,1[} = \varphi_{[0,\frac{1}{2}[} + \varphi_{[\frac{1}{2},1[} \tag{5.2}$$

de forma análoga se define una función dada por la diferencia de dos escalones vecinos de ancho $\frac{1}{2}$ y denotada por $\psi_{[0,1)}$ (wavelet básica), ver figura (5.5)

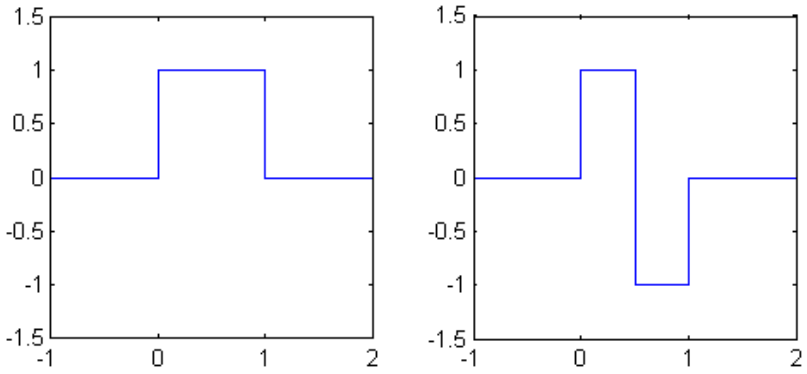


Figura 5.5: La función de escala φ (izquierda) y la wavelet básica ψ (derecha)

$$\begin{aligned} \psi(x) &= \varphi_{[0,1/2[}(x) - \varphi_{[1/2,1[}(x) \\ &= \begin{cases} 1, & x \in [0, \frac{1}{2}) \\ -1, & x \in [\frac{1}{2}, 1) \\ 0, & \text{si no} \end{cases} \end{aligned} \tag{5.3}$$

Entonces sumando y restando $\varphi_{[0,1)}$ y $\psi_{[0,1)}$ obtenemos

$$\varphi_{[0,\frac{1}{2}[} = \frac{1}{2}(\varphi_{[0,1[} + \psi_{[0,1[})$$

$$\varphi_{[\frac{1}{2},1[} = \frac{1}{2}(\varphi_{[0,1[} - \psi_{[0,1[})$$

Lo anterior significa que si se tiene

$$\widetilde{f} = f_0\varphi_{[0,1/2[} + f_1\varphi_{[1/2,1[} \tag{5.4}$$

$$= f_0\frac{1}{2}(\varphi_{[0,1[} + \psi_{[0,1[}) + f_1\frac{1}{2}(\varphi_{[0,1[} - \psi_{[0,1[}) \tag{5.5}$$

$$= \frac{f_0 + f_1}{2}\varphi_{[0,1[} + \frac{f_0 - f_1}{2}\psi_{[0,1[} \tag{5.6}$$

Como se observa la transformada básica preserva toda la información de la muestra, aunque describa la muestra de forma diferente a como lo hacen los valores muestrales. Vamos a multiplicar cada sumando por $\sqrt{2}$ y más adelante (próxima conferencia) vamos a justificar por qué. Tomando en el ejemplo inicial $(5, 1, 2, 8)$ como vector de datos

$$\widetilde{f} = 5\varphi_{[0,1/4[} + 1\varphi_{[1/4,1/2[} + 2\varphi_{[1/2,3/4[} + 8\varphi_{[3/4,1[}$$

y aplicando la transformada básica tendríamos

$$5\varphi_{[0,1/4[} + 1\varphi_{[1/4,1/2[} = \frac{5+1}{2}\varphi_{[0,1/2[} + \frac{5-1}{2}\psi_{[0,1/2[}$$

$$2\varphi_{[1/2,3/4[} + 8\varphi_{[3/4,1[} = \frac{2+8}{2}\varphi_{[1/2,1[} + \frac{2-8}{2}\psi_{[1/2,1[}$$

Sustituyendo en la expresión de $\widetilde{f}$ y multiplicando por $\sqrt{2}$

$$\widetilde{f} = 3\sqrt{2}\varphi_{[0,1/2[} + 2\sqrt{2}\psi_{[0,1/2[} + 5\sqrt{2}\varphi_{[1/2,1[} - 3\sqrt{2}\psi_{[1/2,1[}$$

donde el término $2\psi_{[0,1/2[}$ indica el salto entre 5 y 1 (sería (2)(-2)) y el término $-3\psi_{[1/2,1[}$ indica el salto entre 2 y 8 (sería (-3)(-2)).

La representación anterior significa que separamos la señal en dos bandas de frecuencia, las bajas son las que se obtienen de sumar (2 componentes) y las altas que se obtienen de restar (otras dos componentes) y ahora nos quedamos con la mitad de la muestra, por lo que se reduce la resolución en el tiempo (o lo que es lo mismo aumentamos la escala (la duplicamos); sin embargo se duplica la resolución en frecuencia, ya que sólo nos quedamos con una banda de frecuencias que es la mitad de la original (se reduce la incertidumbre de frecuencia a la mitad). Representamos la función con una resolución menor. Realizando un paso más pues se tiene

$$\widetilde{f} = 4(\sqrt{2})^2\varphi_{[0,1[} - (\sqrt{2})^2\psi_{[0,1[} + 2\sqrt{2}\psi_{[0,1/2[} - 3\sqrt{2}\psi_{[1/2,1[}$$

Antes de formalizar el algoritmo de la transformada wavelet discreta de Haar, veamos un pequeño ejemplo de Fourier contra wavelet.

5.1.1. Fourier vs Wavelets. Un ejemplo

Dada

$$f(t) = 1/4 - 1/2cos\pi t/2 + 1/4cos\pi t,$$

tomemos una muestra de $N = 4$ puntos para $t = 0 : 4$, es decir $f = [0, 0, 1, 0]$. Si calculamos la transformada discreta de Fourier $\widehat{f}(j) = \sum_{l=0}^{3} f(t_l) e^{\frac{-2\pi i j l}{4}}$ obtenemos $\widehat{f} = [1/4, -1/4, 1/4, -1/4]$. Si consideramos la función de escala de Haar $\varphi(t) = \varphi_{[0,1[}(t)$, la representación de Haar en el intervalo $[0, 4]$ será

$$\widetilde{f}(t) = 0\varphi(t) + 0\varphi(t-1) + 1\varphi(t-2) + 0\varphi(t-3).$$

Las aproximaciones de Fourier y de Haar se representan en la figura (**??**).

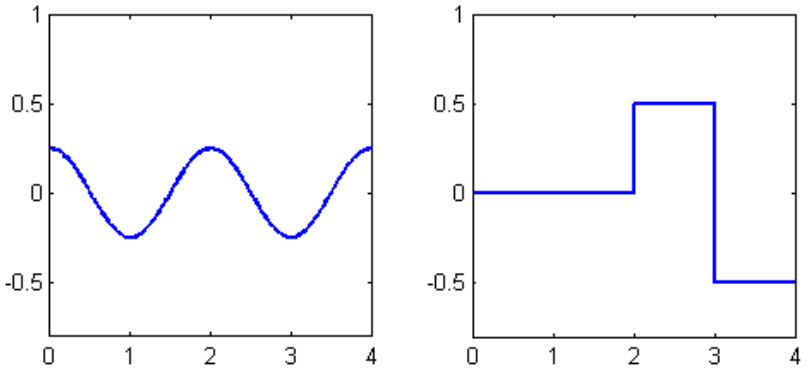


Figura 5.6: Componente de Fourier (izquierda) y Componente de Haar (derecha)

Si realizamos el primer paso de la transformada wavelet de Haar (obviando la división por $\sqrt{2}$)

$$\widetilde{\widetilde{f}}(t) = 0\varphi(t/2) + 0\psi(t/2) + 0{,}5\varphi(t/2-1) + 0{,}5\psi(t/2-1),$$

la componente de la wavelet de Haar con la frecuencia más alta se obtiene de calcular la diferencia entre pares de elementos vecinos, que resulta

$$0\psi(t/2) + 0{,}5\psi(t/2-1).$$

Graficando la componente de la frecuencia más alta en la representación de Fourier y en la wavelet de Haar, ver figura (5.6) se observa que la localización en la representación de Haar es clara y no así en la de Fourier.

5.1.2. Traslaciones y dilataciones de la transformada de Haar básica

Para aplicar la transformada partiendo de un valor $x \neq 0$ y sobre un intervalo que se extiende a w en lugar de 1 se usan las siguientes propiedades. Para todo número x, se cumple que

$$\varphi_{[0,w[}(x) = \varphi_{[0,1)}(\frac{x}{w})$$
$$\varphi_{[u,w[}(x) = \varphi_{[0,1[}(\frac{(x-u)}{w-u})$$

y entonces la función de escala de Haar (5.2) y la wavelet básica de Haar (5.3) se pueden expresar mediante las llamadas relaciones de escala fundamentales

$$\begin{aligned} \varphi_{[0,1[}(x) &= \varphi(2x) + \varphi(2x-1) \\ \psi_{[0,1[}(x) &= \varphi(2x) - \varphi(2x-1). \end{aligned} \tag{5.7}$$

Como se observa la función en una escala se representa mediante la función de escala φ en la escala dividida a la mitad. Si ahora se considera $\Omega = [0,1], j \geq 0$ y se definen los intervalos $\Omega_j^i = [2^{-j}i, 2^{-j}(i+1)[$, para $i = 0, 1, 2, ..., 2^j - 1$; entonces se puede definir la función característica

$$x \rightarrow \varphi(2^j x - i)$$

para el intervalo Ω_j^i y $\forall x \in \Omega$ se aproxima la función f por su proyección $P_j f$ en la familia de funciones constantes en los intervalos Ω_j^i

$$P_j f(x) = \sum_{i=0}^{2^j-1} P_j^i f \varphi(2^j x - i)$$

Como Ω es un dominio acotado y $f \in L^2(\Omega)$, entonces $P_j f \in V_j$,

$$V_j = \left\{ f \in L^2(\Omega), f_{\Omega_j^i} \text{ constante}, \, i = 0, 1, ..., 2^j - 1 \right.$$

V_j tiene dimensión finita, $dim V_j = 2^j$. Entonces se definen las llamadas funciones de escala φ_j^i

$$\varphi_j^i(x) = \varphi\left(2^j x - i\right) = \begin{cases} 1 \text{ si } 2^j x - i \in [0,1) \\ 0 \qquad\qquad \text{si no} \end{cases}$$

Las $n = 2^j$ funciones φ_j^i de altura 1, y constantes por tramos de longitud $h = \frac{1}{2^j}$, forman la llamada **base de Haar**

$$\varphi(x) = \left\{ \varphi_j^i(x), j \in \mathbb{N} \text{ fijo}, n = 2^j, 0 \leq i \leq n-1, \right\},$$

las cuales generan el espacio V_j y son ortonormales relativas al producto escalar en L^2. La ortogonalidad es trivial debido a que estas funciones son distintas de cero en conjuntos disjuntos, queda de ejercicio al lector demostrarlo. La ortonormalidad exige que

$$\int_\Omega \varphi_j^i(x) \, \varphi_j^i(x) \, dx = 1,$$

pero

$$\int_{\Omega} \varphi_j^i(x)\,\varphi_j^i(x)\,dx = \int_{\Omega_j^i} \varphi_j^i(x)\,\varphi_j^i(x)\,dx \tag{5.8}$$

$$= \int_{2^{-j}i}^{2^{-j}(i+1)} dx$$

$$= 2^{-j}(i+1) - 2^{-j}i \tag{5.9}$$

$$= 2^{-j}$$

Por eso entonces para que la base sea ortonormal se considera

$$\varphi_j^i(x) = 2^{\frac{j}{2}}\varphi\left(2^j x - i\right)$$

y podemos escribir $\forall x \in \Omega$

$$P_j f(x) = \sum_{i=0}^{2^j-1} \left\langle f, \varphi_j^i \right\rangle \varphi_j^i(x). \tag{5.10}$$

Los coeficientes $c_j^i = \left\langle f, \varphi_j^i \right\rangle$ son las componentes de $P_j f$ en la base $\left\{\varphi_j^i\right\}$,

$$\left\langle f, \varphi_j^i \right\rangle = \int_{\Omega} f(t)\varphi_j^i dt \tag{5.11}$$

$$= \int_{\Omega_j^i} f(t)\varphi_j^i dt$$

$$= \int_{\Omega_j^i} f(t)dt \tag{5.12}$$

Como

$$2^j x - i = 2^j\left(x - \frac{i}{2^j}\right) = \frac{\left(x - \frac{i}{2^j}\right)}{\frac{1}{2^j}}$$

se dice que la función $\varphi\left(2^j x - i\right)$ está afectada por la traslación $t = \frac{i}{2^j}$ y el escalamiento $s = \frac{1}{2^j}$. Si en nuestro ejemplo introducimos la notación anterior pues se tiene que inicialmente

$$\widetilde{f} = 5\varphi_2^0(x) + 1\varphi_2^1(x) + 2\varphi_2^2(x) + 8\varphi_2^3(x)$$

Si ahora se considera para un entero positivo $j \geq 0$ fijo, los espacios V_j, V_{j+1}, entonces de la propiedad (5.7) se tiene que

$$\varphi_j^i(x) = 2^{\frac{j}{2}}\varphi(2^j x - i) \tag{5.13}$$

$$= 2^{\frac{j}{2}}\varphi\left(2^{j+1}x - 2i\right) + 2^{\frac{j}{2}}\varphi\left(2^{j+1}x - 2i - 1\right)$$

$$= 2^{\frac{j}{2}}\varphi\left(2^{j+1}x - 2i\right) + 2^{\frac{j}{2}}\varphi\left(2^{j+1}x - (2i+1)\right) \tag{5.14}$$

$$= 2^{\frac{j}{2}}2^{\frac{-(j+1)}{2}}\varphi_{j+1}^{2i} + 2^{\frac{j}{2}}2^{\frac{-(j+1)}{2}}\varphi_{j+1}^{2i+1} \tag{5.15}$$

$$= \frac{\varphi_{j+1}^{2i}}{\sqrt{2}} + \frac{\varphi_{j+1}^{2i+1}}{\sqrt{2}}$$

Lo cual implica que $\forall f \in L^2(\Omega), i = 0, .., 2^j - 1$

$$\sqrt{2}\int f(t)\varphi_j^i(t)dt = \int f(t)\varphi_{j+1}^{2i}(t)dt + \int f(t)\varphi_{j+1}^{2i+1}(t)dt$$

De ahí que

$$c_j^i = \frac{c_{j+1}^{2i} + c_{j+1}^{2i+1}}{\sqrt{2}} \tag{5.16}$$

para $i = 0, .., 2^j - 1$.

Entonces los espacios V_j generados por las funciones base ortonormales $\varphi_j^i(x)$ se llaman espacios de aproximación para la escala 2^{-j}

5.1.3. Descomposición del espacio V_J

Sea $J \geq 0$ un entero fijo arbitrario, entonces para cualquier entero j, $0 \leq j \leq J$ se definen los espacios funcionales sucesivos $V_j, V_{j+1}, ..., V_J$, para los cuales se cumple $V_j \subset V_{j+1} \subset ... \subset V_J$. Para cualquier función $f \in L^2(\Omega)$ una forma estándar de escribir la proyección de f en el subespacio V_{j+1} es considerar $P_{j+1}f$ como la proyección ortogonal de f en V_j con un término de corrección

$$\begin{aligned} P_{j+1}f =& P_j f + (P_{j+1}f - P_j f) \\ =& P_j f + Q_j f \end{aligned} \tag{5.17}$$

La relación anterior introduce un nuevo operador $Q_j = P_{j+1} - P_j$, , que es de hecho el operador de proyección ortogonal en W_j, que es a su vez el complemento ortogonal de V_j en V_{j+1};

$$V_{j+1} = V_j \oplus W_j.$$

Es fácil verificar que también se cumple

$$\psi_{[0,1)}(x) = \varphi(2x) - \varphi(2x - 1) \tag{5.18}$$

De forma análoga se define el conjunto de funciones linealmente independientes y de soporte compacto $\Psi(x)$,

$$\Psi(x) = \left\{\psi_j^i(x), j \in \mathbb{N} \text{ fijo}, n = 2^j, 0 \leq i \leq n-1\right\}$$
$$\psi_j^i(x) = \psi\left(2^j x - i\right) = \begin{cases} 1 \text{ si } 2^j x - i \in [0, 1/2) \\ -1 \text{ si } 2^j x - i \in [1/2, 1) \\ 0 \quad \text{si no} \end{cases}$$

Las $n = 2^j$ funciones ψ_i^j forman la **base wavelet de Haar**. Dichas funciones generan el espacio W_j y son ortonormales relativas al producto escalar en L^2.

Para obtener la norma unitaria, realizando un razonamiento análogo al que se hizo para las funciones de escala, se obtiene

$$\psi_j^i(x) = 2^{\frac{j}{2}}\psi\left(2^j x - i\right) \tag{5.19}$$

y aplicando igualmente (5.13),

$$\sqrt{2}\int f(t)\varphi_j^i(t)dt = \int f(t)\varphi_{j+1}^{2i}(t)dt - \int f(t)\varphi_{j+1}^{2i+1}(t)dt$$

para una función cualquiera $f \in L^2(\Omega)$ se pueden calcular los coeficientes d_j^i,

$$d_j^i = \frac{c_{j+1}^{2i} - c_{j+1}^{2i+1}}{\sqrt{2}} \tag{5.20}$$

El coeficiente d_j^i es la fluctuación de f en el intervalo Ω_j^i. Sumando y restando (5.16) y (5.20) se tiene

$$\begin{aligned} \sqrt{2}c_{j+1}^{2i} =& c_j^i + d_j^i \\ \sqrt{2}c_{j+1}^{2i+1} =& c_j^i - d_j^i, \text{ para } i = 0, 1, ., 2^i - 1 \end{aligned} \tag{5.21}$$

respectivamente.

Las expresiones anteriores están directamente relacionadas con la descomposición del espacio V_{j+1} en suma directa $V_{j+1} = V_j \oplus W_j$ y constituyen las relaciones básicas de los algoritmos de descomposición y reconstrucción que formalizaremos posteriormente. Iterando el proceso de descomposición como sigue

$$\begin{aligned} V_J =& V_{J-1} \oplus W_{J-1} \\ =& V_{J-2} \oplus W_{J-2} \oplus W_{J-1} \\ =& \dots \\ =& V_0 \oplus W_0 \oplus \dots W_{J-2} \oplus W_{J-1} \end{aligned} \tag{5.22}$$

Como las funciones φ_j^i (ψ_j^i) forman una base ortonormal de V_j (W_j), entonces estamos en condiciones de definir muchas bases ortonormales de V_j. Dentro de todas se pondrá especial énfasis en dos casos particulares: la base de Haar, llamada en algunos textos base canónica, generada por las funciones φ_j^i, ver desarrollo (5.10) y la llamada base wavelet de Haar o a veces simplemente base de Haar, expandida por φ_0^0 y todas las funciones ψ_j^i, para $j = 0, 1, .., J-1$ y $i = 0, 1, ..., 2^j - 1$. La representación de una función en serie de Haar viene dada entonces por:

$$f_J(x) = c_0^0 + \sum_{j=0}^{J-1}\sum_{i=0}^{2^j-1} d_{ji}\psi_j^i \tag{5.23}$$

Retomando el ejemplo inicial, luego de una primera descomposición sin normalizar se tiene

$$\widetilde{f} = 3\varphi_1^0(x) + 2\psi_1^0(x) + 5\varphi_1^1(x) - 3\psi_1^1(x)$$

Ahora bien para llegar a la representación final en término de las funciones wavelets, aplicando una vez más la transformación sin normalizar se obtiene

$$\widetilde{f} = 4\varphi_0^0(x) + (-1)\psi_0^0(x) + 2\psi_1^0(x) - 3\psi_1^1(x)$$

Como se observa se obtiene una aproximación de la función en un nivel bajo de resolución, expresada por una función de escala y las correspondientes funciones wavelets. El producto del coeficiente

c_j^i por la variación de la función wavelet ψ_j^i (que es fija e igual a -2), nos da la variación que se produce en la función original en el nivel de resolución 2^j en la posición $i/2^j$.
La base de Haar tiene como defecto que está constituida por funciones escalonadas, por lo cual no son buenas para aproximar funciones continuas y mucho menos diferenciables, lo que contribuyó a que fueran olvidadas por mucho tiempo. El algoritmo descrito anteriormente se conoce como **Transformada wavelet discreta de Haar**.

Algoritmo de descomposición
para $j = J-1, ..., 1, 0$ calcular
para $k = 0, 1, ..., 2^j - 1$ calcular
$c_k^j = \left(c_{2k}^{j+1} + c_{2k+1}^{j+1}\right)/\sqrt{2}$
$d_k^j = \left(c_{2k}^{j+1} - c_{2k+1}^{j+1}\right)/\sqrt{2}$
fin
fin

El costo computacional en cada paso es el de calcular 2^{j-1} sumas complejas y 2^{j-1} restas complejas es decir 2^j operaciones complejas

Algoritmo de reconstrucción
para $j = 0, ..., J-1$ calcular
para $k = 0, 1, ..., 2^j - 1$ calcular
$c_{2k}^{j+1} = \left(c_k^j + d_k^j\right)/\sqrt{2}$
$c_{2k+1}^{j+1} = \left(c_k^j - d_k^j\right)/\sqrt{2}$
fin
fin