Matemática Numérica II

Angela León Mecías

Notas de clases, 2014

Índice general

Preface

This is the preface. It is an unnumbered chapter and since it appears before the first numbered chapter the page numbers are typeset in lower case Roman. The preface does not appear in the table of contents.

Capítulo 1

Aproximación de funciones por interpolación

Introducción

Por qué aproximar funciones?

La necesidad de buenas técnicas de aproximación surge en variados marcos de la resolución de problemas reales, como pueden ser: encontrar la solución de problemas de ecuaciones diferenciales, representar curvas, como la que representa el contorno de la bahía de La Habana , figura (1.1) o la

Figura 1.1: Bahía de La Habana

que modela el contorno de una pieza, figura (1.2)

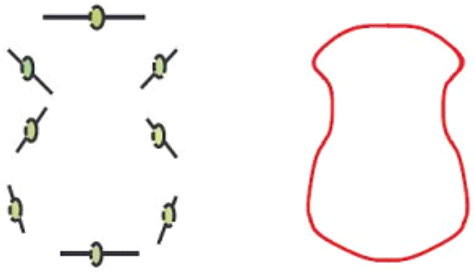

Figura 1.2: Diseño de Pieza

En un contexto más matemático un problema tipo puede ser: conocida una función f de forma exacta o aproximada, 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 con la nueva función que se propone como aproximación de la dada, debe ser mas fácil realizar operaciones tales como diferenciación e integración.

En general no se tiene una expresión analítica de la función conocida si no que se dispone de una tabla de valores proveniente de experimentos o mediciones. A continuación relacionamos algunas de las situaciones más comunes 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 analítica de f y ésta resulte complicada para hacer operaciones tales como diferenciación o integración. En este caso se usa la **aproximación por interpolación**, ver la figura 1.3

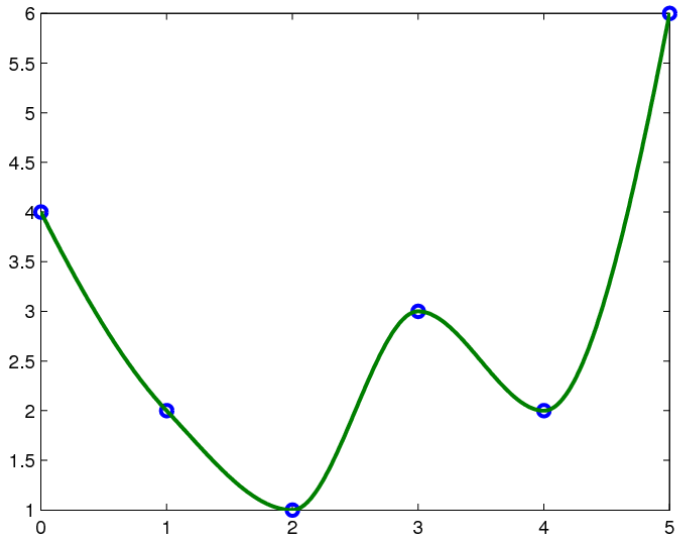

Figura 1.3: Función que interpola los datos

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 $f(x)$ por una suma finita

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

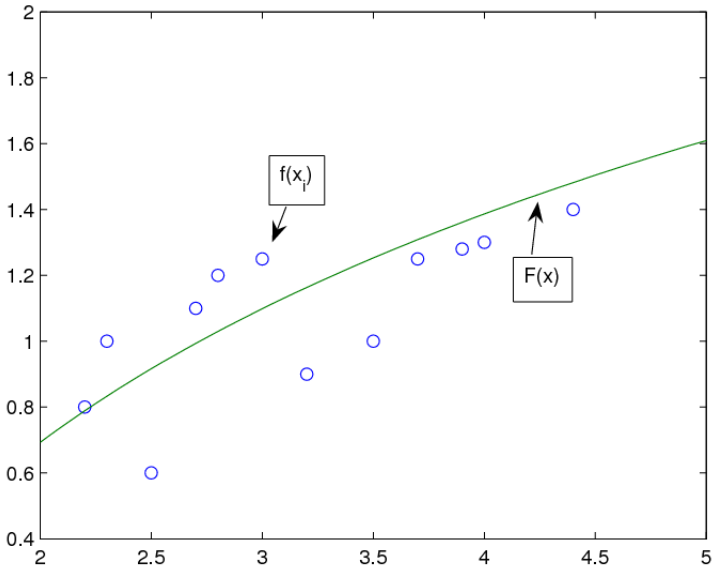

Figura 1.4: Aproximación por mínimos cuadrados

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[1] en 1822, a Euler[2] y Lagrange[3] 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 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

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

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

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

considera un problema inverso, y en el caso en que y es la incógnita, estamos ante un problema directo, como 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 en 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 (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.

Sea $f(x) \in C[a,b]$ dada por una tabla de $n+1$ pares de puntos (n 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, éstos deben ser determinados de forma tal que para los $n+1$ pares de números reales ó complejos

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

se cumple

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

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 $a_0, a_1, \ldots, a_n$

$$\begin{array}{rcll} i & = & 0, & a_0\varphi_0\left(x_0\right) + a_1\varphi_1\left(x_0\right) + \ldots + a_n\varphi_n\left(x_0\right) = f\left(x_0\right) \\ i & = & 1, & a_0\varphi_0\left(x_1\right) + a_1\varphi_1\left(x_1\right) + \ldots + a_n\varphi_n\left(x_1\right) = f\left(x_1\right) \\ & & \vdots & \\ i & = & n, & a_0\varphi_0\left(x_n\right) + a_1\varphi_1\left(x_n\right) + \ldots + a_n\varphi_n\left(x_n\right) = f\left(x_n\right) \end{array} \tag{1.3}$$

Este sistema tendrá solución única si y solo si

$$\det \begin{bmatrix} \varphi_0\left(x_0\right) & \varphi_1\left(x_0\right) \ldots \varphi_n\left(x_0\right) \\ \varphi_0\left(x_1\right) & \varphi_1\left(x_1\right) \ldots \varphi_n\left(x_1\right) \\ & \vdots & \\ \varphi_0\left(x_n\right) & \varphi_1\left(x_n\right) \ldots \varphi_n\left(x_n\right) \end{bmatrix} \neq 0 \tag{1.4}$$

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

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

$$p_n(x_i) = f(x_i) \tag{1.5}$$

Demostración 2 *Unicidad: supongamos que existen dos polinomios $p(x)$ y $q(x)$ de grado menor ó igual que n que cumplen la exigencia de interpolación (1.5). 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 (1.5). 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

$$p_n(x) = \sum_{j=0}^{n} a_j \varphi_j(x) = \sum_{j=0}^{n} a_j x^j \tag{1.6}$$

e imponiendo la exigencia de interpolación, se obtiene

$$p_n(x_i) = \sum_{j=0}^{n} a_j x_i^j = f(x_i), \quad 0 \leq i \leq n \tag{1.7}$$

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

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots x_0^n \\ 1 & x_1 & x_1^2 & \dots x_1^n \\ & & \dots & \\ 1 & x_n & x_n^2 & \dots x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \tag{1.8}$$

Esto es $Va = f$. Si los nodos de interpolación son todos distintos, $\neq 0$, 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,

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

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 (1.8), que resulta de usar la base

$$\left\{1, x, x^2, ..., x^n\right\} \tag{1.9}$$

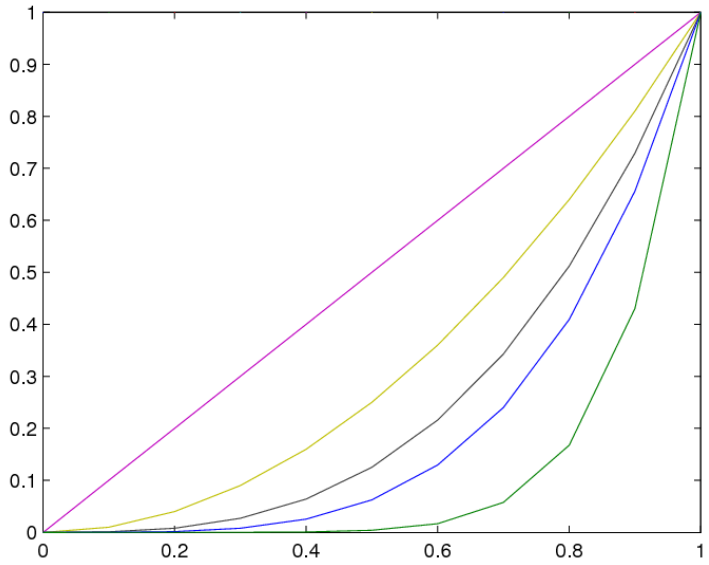

Figura 1.5: Funciones de la base $\{1, x, x^2, ..., x^n\}$

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 partiendpo del punto $(0, 0)$ hasta el $(1, 1)$ cuando $n > 0$:

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 (1.9), 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

$$l_j(x) = \frac{(x - x_0)(x - x_1)\dots(x - x_{j-1})(x - x_{j+1})\dots(x - x_n)}{(x_j - x_0)(x_j - x_1)\dots(x_j - x_{j-1})(x_j - x_{j+1})\dots(x_j - x_n)} \tag{1.10}$$

que cumplen

$$l_j(x_i) = \begin{cases} 1, & \text{si} \quad i = j \\ 0, & \text{si} \quad i \neq j \end{cases} \tag{1.11}$$

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 $(f, g) := \int_a^b f(x)g(x)dx$, es decir se verifica $(l_j, l_k) = 0$, $j \neq k$). Al usar la base de Lagrange, el polinomio de interpolación

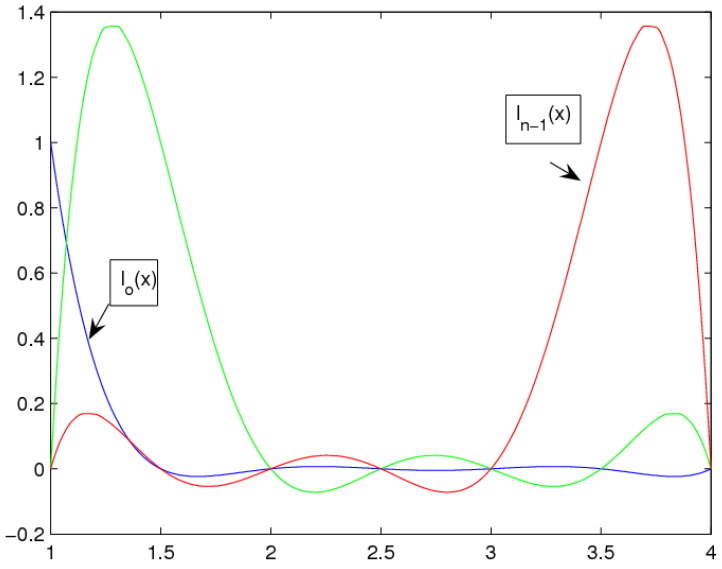

Figura 1.6: Funciones base de Lagrange

tendrá la forma

$$p_n(x) = \sum_{j=0}^{n} a_j l_j(x) \tag{1.12}$$

y la exigencia de interpolación

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

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 $i = j$,

$$a_j l_j(x_j) = f(x_j) \Rightarrow a_j = f(x_j), \quad 0 \leq j \leq n \tag{1.13}$$

Luego sustituyendo (1.13) en (1.12), se obtiene la **fórmula de Lagrange** para el polinomio de interpolación de grado $\leq n$ con nodos $x_0, x_1, ..., x_n$

$$\varphi(x) = p_n(x) = \sum_{j=0}^{n} f(x_j) l_j(x), \quad l_j(x) = \prod_{\substack{i=0 \\ i \neq j}} \left(\frac{x - x_i}{x_j - x_i} \right) \tag{1.14}$$

El uso de la base de Lagrange ha permitido obtener los coeficientes a_j del polinomio sin tener que resolver el sistema (1.3), 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 gráfico en computadora (CAD).(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

$$\varphi(x) \equiv p_3(x) = \sum_{i=0}^{3} f(x_j)\, l_j(x) \tag{1.15}$$

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

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

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

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

Sustituyendo en (1.15) 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 quisiéramos 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 CAD (Çomputed assited design ")

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

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

- 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')

Objeciones a la fórmula de Lagrange

En la práctica, no siempre se conoce apriori cuántos nodos de interpolación deben usarse para lograr una cierta precisión de $p_n(z)$. Entonces, si denotamos por $p_i(x)$ el polinomio de grado $\leq i$ 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 $f[x_0, .., x_{n-1}]$:

$$f[x_0, .., x_n] = \frac{f[x_1, .., x_n] - f[x_0, .., x_{n-1}]}{x_n - x_0} \tag{1.16}$$
$$= \sum_{i=0}^{n} \frac{f(x_i)}{(x_i - x_0) ... (x_i - x_{i-1}) (x_i - x_{i+1}) ... (x_i - x_n)}$$

La tabla de diferencias divididas

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

$$f[x_j, .., x_{j+k}] = \frac{f[x_{j+1}, .., x_{j+k}] - f[x_j, .., x_{j+k-1}]}{x_{j+k} - x_j} \tag{1.17}$$

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 manual de la tabla se efectúa por columnas, usando la fórmula recursiva (1.17) para $k = 1, 2, ..,$, es decir, primero se calculan todas las primeras diferencias divididas, después todas las segundas diferencias divididas, y así sucesivamente

x_0	$f[x_0]$			
x_1	$f[x_1]$	$f[x_0, x_1]$		
x_2	$f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$	
	$f[x_3]$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

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

Deducción de la fórmula de Newton[4]

Partimos del planteamiento

$$p_n(x) = p_{n-1}(x) + h(x) \tag{1.18}$$

[4] 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

$$h(x) = a_n (x - x_0)(x - x_1) ... (x - x_{n-1}) \tag{1.19}$$

donde a_n es una constante a determinar. Sustituyendo (1.19) en (1.18), obtenemos:

$$p_n(x) = p_{n-1}(x) + a_n (x - x_0)(x - x_1) ... (x - x_{n-1}) \tag{1.20}$$

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

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

de donde

$$a_n = \frac{f(x_n) - p_{n-1}(x_n)}{(x_n - x_0) ... (x_n - x_{n-1})} \tag{1.21}$$

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 (1.21) y simplificando se obtiene

$$a_n = \sum_{i=0}^{n} \frac{f(x_i)}{(x_i - x_0) ..(x_i - x_{i-1})(x_i - x_{i+1})...(x_i - x_n)} \tag{1.22}$$

Comparando (1.22) 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

$$a_n = f[x_0, x_1, ..., x_n] \tag{1.23}$$

y sustituyendo (1.23) en (1.20), obtenemos

$$p_n(x) = p_{n-1}(x) + f[x_0, x_1, ..., x_n](x - x_0)(x - x_1) ... (x - x_{n-1}) \tag{1.24}$$

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

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

$$f[x_0] = f(x_0) \tag{1.25}$$

lo que podemos tomar como definición de la diferencia dividida de orden 0 (cero) de f en x_0.

Teniendo en cuenta (1.24) y (1.25), obtenemos finalmente

$$\begin{aligned} p_n(x) = {} & f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ & + .. + f[x_0, x_1, .., x_n](x - x_0)..(x - x_{n-1}) \end{aligned} \tag{1.26}$$

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

$$p_n(x) = \sum_{i=0}^{n} f[x_0, .., x_i] \prod_{j=0}^{i-1} (x - x_j) \tag{1.27}$$

Nótese 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*

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

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 (1.27), 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

Ejemplo 8 .

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

Construcción de las aproximaciones de $f(1)$:

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

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

$p_1(1) = 2$

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

$p_2(1) = 2{,}67$

$n = 3$: $p_3(x^*) = p_2(x^*) + f[x_0, x_1, x_2, x_3](-1)(x^* - x_2)$

$p_3(1) = 3{,}27$

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

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

Demostración 10 *Según (1.27)* $f[x_0, .., x_k]$ *es el coeficiente de la mayor potencia del polinomio de interpolación* $P_{0,...,k}$ *que pasa por los puntos* $(x_i, f_i), i = 0, ..k$*. Teniendo en cuenta que el polinomio de interpolación es único entonces se cumple* $P_{i_0,...,i_k}(x) \equiv P_{0,...k}(x)$ *para una permutación cualquiera* $i_0, ..i_k$ *de los números* $0, 1, ..., k$*. En particular se cumple entonces que*

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

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);

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

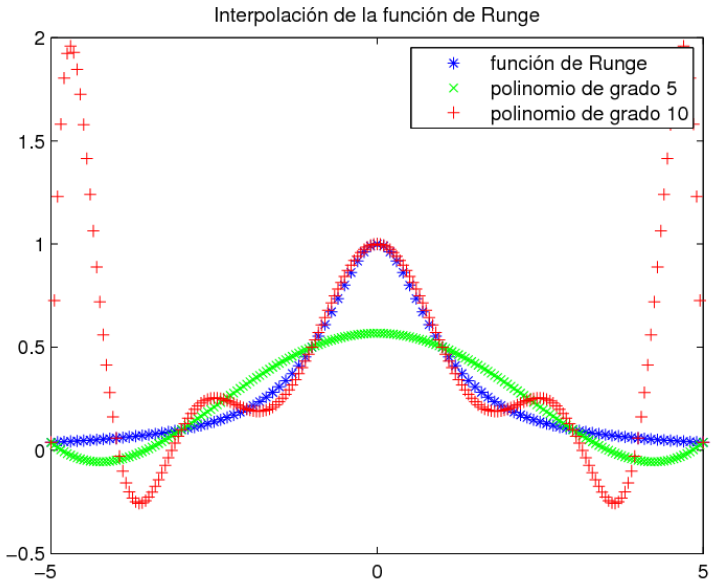

Figura 1.7: Función de Runge

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).

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

$$\psi_i(x) = \prod_{j=0}^{i-1} (x - x_j) = (x - x_0) \ldots (x - x_{i-1}) \tag{1.28}$$

Si conocemos los valores $p_{n-1}(x^*), \psi_{n-1}(x^*)$ y $f[x_0, .., x_n]$, 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 11

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 12 (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 13 *Dada $f \in C[a, b]$ arbitraria, existe $\{x_i\}_{i=0}^n$ tal que $\{P_n(x)\} \to f(x)$ uniformemente.*

Observación 14 *El teorema 12 indica que no hay esquemas de interpolación universales efectivos que tengan como base sólo valores de f. Ejemplo la función de Runge*

Observación 15 *Aunque el teorema 13 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 funciones que se conocían en forma tabular, en puntos intermedios. Cuando en tales tablas los valores f_i están dados 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, que consideraremos a continuación.

El espaciamiento h entre dos valores consecutivos x_i y x_{i+1} de la variable independiente se denomina paso de la tabla.

Introduzcamos el cambio de variable

$$s = s(x) = \frac{x - x_0}{h} \tag{1.29}$$

de donde

$$x = x(s) = x_0 + sh \tag{1.30}$$

Note que en (1.30), si $s \in \aleph$ entonces $x = x_s$ es nodo de interpolación. De acuerdo con(1.30), se tiene que

$$f(x_s) = f(x_0 + sh) \tag{1.31}$$

Como el cambio de variable (1.30) 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 $\leq n$ 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

$$f_{i+1} - f_i = f(x_i + h) - f(x_i) = \begin{cases} \Delta f_i : \text{diferencia finita hacia adelante en el nodo } x_i \\ \nabla f_{i+1} : \text{ diferencia finita hacia atrás en el nodo } x_{i+1} \end{cases} \tag{1.32}$$

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:

$$\Delta^n f_i = \begin{cases} f_i, \text{ para } n = 0 \\ \Delta\left(\Delta^{n-1} f_i\right) = \Delta^{n-1} f_{i+1} - \Delta^{n-1} f_i, \quad \text{para } n > 0 \end{cases} \tag{1.33}$$

y análogamente para las diferencias finitas hacia atrás:

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

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:

$$f\left[x_i, \ldots, x_{i+k}\right] = \frac{1}{k!h^k}\Delta^k f_i \quad , \quad \forall k \geq 0 \tag{1.35}$$

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 :

$$f\left[x_0, ..., x_k\right] = \frac{f^{(k)}(\xi)}{k!} = \frac{\Delta^k f_0}{k!h^k} \quad , \quad x_0 < \xi < x_k \tag{1.36}$$

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 (1.35), obtenemos

$$p_n(x) = \sum_{i=0}^{n} \frac{1}{i!h^i}\Delta^i f_0 \prod_{j=0}^{i-1}(x - x_j) \tag{1.37}$$

b) En términos de s, tenemos que

$$x - x_j = (x_0 + sh) - (x_0 + jh) = h(s - j) \tag{1.38}$$

Sustituyendo (1.38) en (1.37)

$$p_n(x) = p_n(x_0 + sh) = \sum_{i=0}^{n} \frac{1}{i!h^i}\Delta^i f_0 \prod_{j=0}^{i-1} h(s - j) \tag{1.39}$$

$$= \sum_{i=0}^{n} \Delta^i f_0 \frac{s!}{i!(s - i)!} \tag{1.40}$$

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

$$b(y) = \begin{pmatrix} y \\ i \end{pmatrix} = \begin{cases} 1, & \text{si } i = 0 \\ \prod_{j=0}^{i-1} \left(\frac{y-j}{j+1}\right) = \left(\frac{y}{1}\right)\left(\frac{y-1}{2}\right)...\left(\frac{y-i+1}{i}\right), & \text{si } i > 0 \end{cases} \tag{1.41}$$

La palabra binomial se justifica, pues si $y \in \mathbb{N}$ entonces la expresión (1.41) coincide con la del coeficiente binomial:

$$\begin{pmatrix} y \\ i \end{pmatrix} = \frac{y!}{i!\,(y-i)}$$

d) Sustituyendo (1.41) en (1.40) , obtenemos

$$p_n(x_0 + sh) = \sum_{i=0}^{n} \Delta^i f_0 \begin{pmatrix} s \\ i \end{pmatrix}, \; s = \frac{x - x_0}{h} \tag{1.42}$$

que en forma desarrollada nos da

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

y sustituyendo las funciones binomiales,

$$p_n(x_0 + sh) = f_0 + s\Delta f_0 + \frac{s(s-1)}{2!}\Delta^2 f_0 + ... + \frac{s(s-1)...(s-n+1)}{n!}\Delta^n f_0 \tag{1.43}$$

Las epresiones (1.42) y (1.43) reciben el nombre de **fórmula de Newton en diferencias finitas hacia adelante.**

Los coeficientes $\Delta^i f_0$ de (1.42) y (1.43) son los que encabezan las columnas de la tabla de diferencias finitas construida apartir de los nodos $x_0, ..., x_n$.

Aplicaciones

- Cálculo del valor interpolado
- Cálculo de ceros
- Derivación aproximada
- Integración aproximada

Resumen de fórmulas de Newton en diferencias finitas

- Fórmula de Newton en diferencias finitas hacia adelante (ver clase)
- Fórmula de Newton en diferencias finitas retrógradas o hacia atrás

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

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

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

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

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

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

1.2. La forma de Hermite del polinomio de interpolación

Los polinomios de Hermite [5], la forma normal de Hermite y el spline cúbico de Hermite son llamados así en honor al nombre de su descubridor. 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. Tanto para el caso del polinomio de Lagrange como para el polinomio de Newton se parte de una función tabulada, es decir una tabla donde se tienen valores de la función en un conjunto determinado de puntos. ¿Qué pasa si además se tuvieran valores de las derivadas? ¿Cómo se podría aprovechar esta nueva información? A diferencia de Lagrange y de Newton la interpolación por el polinomio de Hermite si aprovecha tal información, de ahí el interés en su estudio. Sean dados los números reales:

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

Se tienen $m+1$ puntos y se conoce hasta la derivada de orden $n_i - 1$, es decir, en cada punto conozco determinadas derivadas.

La interpolación de Hermite con respecto a dichos datos se basa en determinar un polinomio P de grado $\leq n$, $n + 1 := \sum_{i=0}^{m} n_i$ que satisfaga las siguientes condiciones de interpolación,

$$P^k(\xi_i) = y_i^{(k)},\ k = 0, \dots n_i - 1,\ i = 0, \dots m \tag{1.44}$$

A diferencia de la interpolación usual, que se obtiene como caso particular cuando $n_i = 1$, se conoce en cada punto no solo el valor de la función sino de la derivada. Las condiciones (1.44) nos dan exactamente $\sum_{i=0}^{m} n_i = n + 1$ condiciones para los $n + 1$ coeficientes de P, de manera que se espera la unicidad en la solución del problema.

La existencia y unicidad del polinomio de grado n para $m + 1$ puntos con $n + 1 := \sum_{i=0}^{m} n_i$ se demuestra igual que para Lagrange,

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

[5] Charles Hermite (1822-1901) matemático francés que investigó temas como teoría de números, formas cuadráticas, polinomios ortogonales, algebra, etc

con $L_{ik}(x) \in \Pi_n$ polinomios de Lagrange generalizado con

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

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

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

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.

Conclusiones 17 *Incoveniencia del uso 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. 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.

Preguntas de comprobación

Cuál es el objetivo de la función de Newton para la construcción del polinomio de interpolación?

Cómo se usa le fórmula de Newton para la estimación de $f(x^*)$ cuando no se tiene apriori criterio sobre el grado a tomar?

1.3. Interpolación por tramos

Anteriormente se vio cómo construir el polinomio de interpolación por dos vías, usando la fórmula de Lagrange y la de Newton en diferencias finitas hacia adelante, tanto para nodos no equidistantes como equidistantes. En ambos casos interesa poder estimar el error de valores interpolados $p_n(x^*)$ en puntos x^* que no sean nodos. Como se observa en la siguiente figura, cuando se tienen muchos nodos, construir el polinomio de interpolación no brinda una buena aproximación, en este caso se introduce la interpolación por tramos.

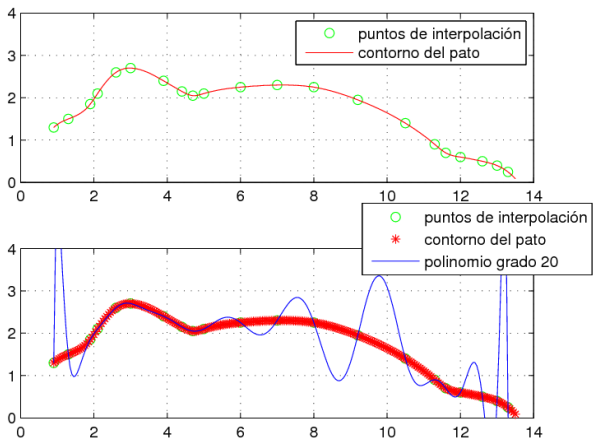

Figura 1.8: Oscilaciones que se presentan al aproximar por polinomios de orden elevado

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 $x_o, x_1, \ldots, x_n$ de I. Sea $p_n(x)$ el polinomio de grado $\leq n$ que interpola a f en los nodos $x_o, \ldots, x_n$; el "error $e_n(x)$ del polinomio de interpolación"se define como:

$$e_n(x) := f(x) - p_n(x). \tag{1.45}$$

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

$$p_{n+1}(x^*) = f(x^*), \tag{1.46}$$

y de acuerdo con la fórmula de Newton

$$p_{n+1}(x) = p_n(x) + f[x_o, \ldots, x_n, x^*](x - x_o) \cdots (x - x_n). \tag{1.47}$$

Luego, de (1.46) y (1.47) obtenemos

$$f(x^*) = p_{n+1}(x^*) = p_n(x^*) + f[x_o, \ldots, x_n, x^*](x^* - x_o) \ldots (x^* - x_n), \tag{1.48}$$

y sustituyendo (1.48) en (1.45),

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

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

$$e_n(x^*) = f[x_o, \ldots, x_n, x^*](x^* - x_o) \cdots (x^* - x_n). \tag{1.49}$$

Esta expresión muestra que el error en el punto x^* es *parecido al próximo término* de la fórmula de Newton, $f[x_o, \ldots, x_n, x_{n+1}](x^* - x_o) \cdots (x^* - x_n)$, 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 (1.49) no puede ser evaluada, a menos que se conozca el valor $f(x^*)$ y con él se calcule $f[x_o, \ldots, x_n, x^*]$. Pero como veremos a continuación, el número $f[x_o, \ldots, x_n, x^*]$ 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 18 *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 $x_o, \ldots, x_n$ en $[a,b]$, entonces para todo $x^* \in [a,b]$, existe $\xi = \xi(x^*) \in (a,b)$ tal que*

$$e_n(x^*) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^{n} (x^* - x_j) \tag{1.50}$$

Demostración 19 *Sea $x^* \neq x_0, .., x_n$ 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 $w(x) = (x-x_0)....(x-x_n)$, entonces $g(t)$ tiene $(n+2)$ ceros en los nodos $I = [x_o, \ldots, x_n, x^*]$. 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

$$f(x^*) - p_n(x^*) = \frac{f^{(n+1)}(\xi)}{(n+1)!} w(x^*) \tag{1.51}$$

Teorema 20 *Sea $f(x)$una función que toma valores reales, continua sobre un intervalo $[a,b]$ y k veces diferenciable en (a,b). Si $x_o, \ldots, x_k$ son $k+1$ puntos distintos en $[a,b]$, entonces existe $\xi \in (a,b)$, tal que*

$$f[x_0, .., x_k] = \frac{f^{(k)}(\xi)}{k!} \tag{1.52}$$

Demostración 21 *Si consideramos (1.48)*

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

tenemos que

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

y comparando con (1.51), se tiene que

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

con lo que se cumple en general

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

Observación 22 *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 (1.50) es de utilidad práctica limitada pues, en general, $f^{(n+1)}(x)$ y el punto ξ se desconocen. Una cota superior grosera de $e_n(x)$ está dada por

$$|e_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \max_{x\in[a,b]} |(x - x_o)\cdots(x - x_n)| \tag{1.53}$$

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 23 *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 (1.50) 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 24

- *$f[x_o, \ldots, x_n, x^*]$ puede estimarse mediante $f[x_o, \ldots, x_n, x_{n+1}]$ 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 (1.50) del error en el punto x^* se puede separar en dos partes:

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

donde la primera depende de f y no se puede alterar y la segunda depende sólo de los nodos. La reducción del error se puede llevar a cabo sólo variando el segundo factor, lo que se puede realizar de varias maneras, por ejemplo,

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 $\prod_{j=o}^{N}(x - x_j)$, 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 $[x_o, x_n]$).

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.

1.3.1. 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).

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

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 :

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

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 $[x_o, x_N]$.

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

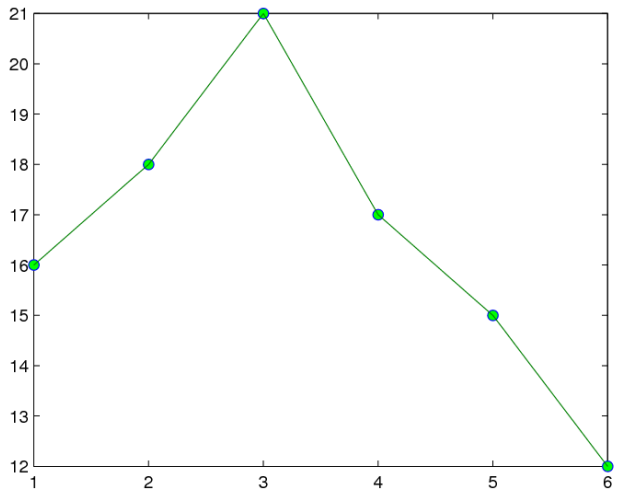

Figura 1.10: Interpolación lineal por tramos

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 lo suficientemente suaves, es decir con un determinado orden de diferenciabilidad.

Definición 25 *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 3er. grado definidos localmente, que posee continuidad global y primera y segunda derivadas continuas globalmente.

Definición 26 *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 $i = 0, 1, ...n-1$, $S(x)$ es un polinomio cúbico, denotado por S_i en $[x_{i-1}, x_i)$.*
- $S(x_i) = f(x_i)$, $i = 0, 1, ...n$
- $S_{i-1}(x_i) = S_i(x_i)$, $i = 0, 1, ...n-2$
- $S'_{i-1}(x_i) = S'_i(x_i)$, $i = 0, 1, ...n-2$
- $S''_{i-1}(x_i) = S''_i(x_i)$, $i = 0, 1, ...n-2$

Además 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 y conocida en el lenguaje común como "la culebra".

Deducción del spline cúbico natural

Sea la función $f : \Re \longrightarrow \Re$ dada mediante la tabla

x	x_o	x_1	x_2	$\cdots$	x_N
f(x)	f_o	f_1	f_2	$\cdots$	f_N

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

$$S_i(x) = \begin{cases} a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \text{ si } x \in [x_{i-1}, x_i] \\ 0, \text{ si } x \notin [x_{i-1}, x_i] \end{cases} \tag{1.54}$$

con $1 \leq i \leq N$.

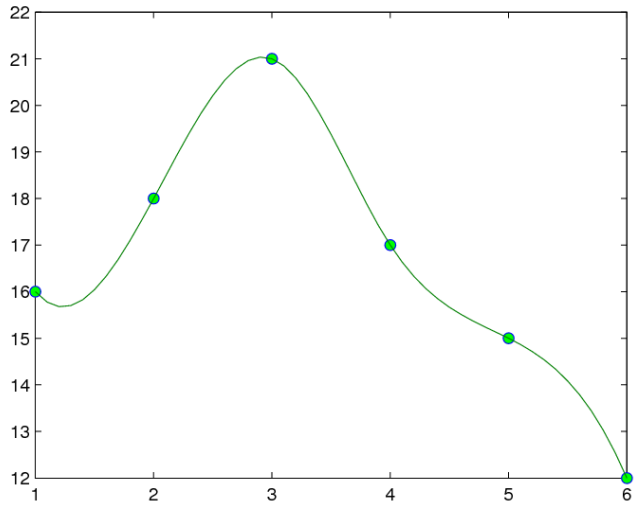

Figura 1.11: Aproximación por tramos con Spline Cúbico

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

$$S_i(x_{i-1}) = f_{i-1} \quad \text{y} \quad S_i(x_i) = f_i, \tag{1.55}$$

en total, $2(N-1)$ en los nodos interiores más 2 en los nodos extremos, o sea, $2N$. Las dos condiciones restantes 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:

$$S_i{}'(x_i) = S'_{i+1}(x_i) \quad \text{y} \quad S\,''_i(x_i) = S\,''_{i+1}(x_i), \tag{1.56}$$

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 (1.54) en el intervalo $[x_{i-1}, x_i]$

$$S'_i(x) = 3a_i(x-x_i)^2 + 2b_i(x-x_i) + c_i \tag{1.57}$$

$$S\,''_i(x) = 6a_i(x-x_i) + 2b_i \tag{1.58}$$

evaluando $S_i(x)$ en $x = x_i$, $S_i(x_i) = d_i$, y teniendo en cuenta (1.55), se tiene

$$d_i = f_i. \tag{1.59}$$

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,

$$b_i = \frac{t_i}{2} \quad \text{y} \quad a_i = \frac{t_i - t_{i-1}}{6(x_i - x_{i-1})}. \tag{1.60}$$

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

$$c_i = \frac{f_i - f_{i-1}}{h_i} + \left(\frac{2t_i + t_{i-1}}{6}\right) h_i. \tag{1.61}$$

Las expresiones (1.59), (1.60) y (1.61) 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 (1.56) en $x_i, (1 \le i \le N)$:

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

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

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 (1.56) 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

$$\begin{aligned} & h_i t_{i-1} + 2(h_i + h_{i+1})t_i + h_{i+1}t_{i+1} \\ = \; & 6\left(\frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i}\right), \; (1 \le i \le N-1). \end{aligned} \tag{1.62}$$

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

$$t_o = S_0''(x_0) = t_N = S_N''(x_N) = 0, \tag{1.63}$$

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 (1.62)-(1.63) representado matricialmente será:

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

$$= 6 \begin{bmatrix} (f[x_1, x_2] - f[x_o, x_1]) \\ (f[x_2, x_3] - f[x_1, x_2]) \\ \cdots \\ (f[x_{N-2}, x_{N-1}] - f[x_{N-3}, x_{N-2}]) \\ (f[x_{N-1}, x_N] - f[x_{N-2}, x_{N-1}]) \end{bmatrix}. \tag{1.64}$$

donde $f[x_{i-1}, x_i] = \frac{f_i - f_{i-1}}{x_i - x_{i-1}}$. La matriz del sistema (1.64) es definida positiva, lo que garantiza solución única. Dicha matriz es simétrica y tridiagonal, por lo que resulta adecuada la aplicación del método de Choleski 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 $t_o = t_N = 0$, en las expresiones (1.59), (1.60) y (1.61) 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 27 *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 $\sqrt{55} = 7{,}416$.

1.3.2. 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 $x_k \leq x \leq x_{k+1}$,

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

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

$$\begin{aligned} P(x_k) &= y_k,\ P(x_{k+1}) = y_{k+1}, \\ P^{'}(x_k) &= d_k,\ P^{'}(x_{k+1}) = y_{k+1} \end{aligned} \tag{1.65}$$

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 (1.65), 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.

1.4. 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
y	2	1	6	47

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

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

2. Utilizando la función vander de Matlab y la división izquierda encuentre los coeficientes del polinomio de interpolación para los datos: $x = 0 : 10 ey = 1 : 11$. Obtenga mediante el programa Lagrainterp la expresión analítica del polinomio de Lagrange, haciendo uso de los comandos simbólicos de Matlab **sym** y **pretty**

$$\begin{aligned} symx &= sym('x') \\ p &= Lagrainterp(x, y, symx); \\ & pretty(p); \\ p &= simplify(p); \end{aligned} \tag{1.66}$$

   Notar como el polinomio que resulta de utilizar el comando **vander** difiere del que se obtiene mediante **Lagrainterp**. ¿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

   Utilice la interpolación de Hermite para predecir cuánto había recorrido la motocicleta después de 10s y cuál era su velocidad en ese momento. ¿Cómo valora los resultados obtenidos? ¿Serán una buena aproximación?

Capítulo 2

Aplicaciones de la interpolación

Anteriormente estudiamos cómo aproximar una función por un polinomio de interpolación. En muchos casos la derivación y la integración de funciones no puede realizarse exactamente mediante los métodos del cálculo diferencial e integral y hay que recurrir a la resolución aproximada de dichos problemas. Esto puede lograrse mediante otro empleo del polinomio de interpolación, de mayor importancia en la práctica que el de hallar valores aproximados de una función f. 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. Diferenciación numérica

La diferenciación numérica la utilizamos cuando la función f está dada por una tabla o cuando su expresión analítica es muy compleja. En el primer caso no es posible aplicar los métodos del Cálculo Diferencial, en el segundo su utilización origina dificultades considerables.

Para deducir fórmulas para la diferenciación aproximada, sustituimos f definida sobre el intervalo $[a,b]$ por un polinomio de interpolación $p(x)$ y derivamos,

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

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 diferenciación 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,

$$P_n(x_0 + sh) = f_0 + s\Delta f_0 + \frac{s(s-1)}{2!}\Delta^2 f_0 + ... + \frac{s(s-1)...(s-n+1)}{n!}\Delta^n f_0 \qquad (2.1)$$

Además teníamos la siguiente relación:

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

Entonces si aproximamos la función f por (2.1), 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, como se dijo anteriormente, por la derivada del error de interpolación, es decir por el término $n+1$ de la expresión correspondiente.

Ejemplo 28 *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 de f' puede obtenerse también a partir del desarrollo en serie de Taylor, truncando después del segundo término y dividiendo por h.

Ejemplo 29 *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 tmbién 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 30 *Aproximar la primera derivada en $x=x_0$ con $n=2$, con nodos de interpolación x_0, x_1, x_2*

$x = x_0 \curvearrowright s = 0$

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

$E\left(f'\left(x\right)\right) \curvearrowright E_2\left(f'\right) \approx \frac{1}{h}\left[\frac{3s^2-6s+2}{6}\Delta^3 f_0\right] = \frac{h^2}{3}f'''\left(\xi\right) = O\left(h^2\right)$ *Nótese que en este caso la vía de Taylor no es fácil de adivinar, sin embargo la función de Newton nos da un método general para aproximar derivadas.*

2.2. Integración aproximada

El problema que nos ocupa es el de tener que calcular la integral de una función $f\left(x\right)$,

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

cuando no es posible hacerlo de manera exacta, lo cual ocurre en la mayoría de los problemas prácticos, ya sea porque:

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. o porque no se tiene una expresión analítica de f, sino sólo 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 $I\left(f\right)$. Intuitivamente lo primero en lo que pensaríamos es en usar la definición de integral definida como límite cuando $n \to \infty$ de la suma integral $S_n = \sum_{i=1}^{n} f\left(x_i\right)\Delta x_i$. Sin embargo la convergencia de S_n a $I\left(f\right)$ es muy lenta, por lo que esta vía no se usa en la práctica. El problema de la integración numérica o cuadratura numérica consiste entonces en estimar el valor de

$$\left(f\right) = \int_a^b f\left(x\right)dx \quad \text{por el de } I\left(p\right) = \int_a^b p\left(x\right)dx$$

En este caso vamos a considerar que la función $f\left(x\right)$ se sustituye por un polinomio de interpolación $p(x)$

$$f \approx p \curvearrowright I\left(f\right) \approx I\left(p\right)$$

Consideremos para ello que f es una función suave en el intervalo $[a,b]$ y que $p_n\left(x\right)$ es el polinomio de interpolación de grado $\leq n$ 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 entonces la expresión dada por la fórmula de Newton en diferencias finitas hacia adelante :

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

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.

2.2.1. Fórmulas de Newton Cotes

Integrando la expresión anterior entre $a = x_0$ y $b = x_n$ (para obtener fórmulas de integración de tipo cerrado, es decir usando los valores de la función del integrando en los extremos del intervalo de integración):

$$\begin{aligned} I(f) &= \int_a^b f(x)\,dx \approx I(p) = \int_{x_0=a}^{x_n=b} p(x)\,dx \\ &= \int_0^n \left\{ f_0 + \Delta f_0 \begin{pmatrix} s \\ 1 \end{pmatrix} + \Delta^2 f_0 \begin{pmatrix} s \\ 2 \end{pmatrix} + ... + \Delta^n f_0 \begin{pmatrix} s \\ n \end{pmatrix} \right\} hds \end{aligned} \tag{2.2}$$

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 por la integral del error de interpolación.:

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

Las fórmulas de integración numérica que se obtienen a partir de la expresión (1) , 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.

En lo que sigue veremos como casos particulares, 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

$n = 1$: Regla básica de los trapecios en $[x_{i-1}, x_i]$ Si los nodos son equidistantes, entonces $h = x_i - x_{i-1} \ \ \forall i$ y se obtiene:

$$\begin{aligned} I(p_1) &= \int_0^1 \left[f_{i-1} + \Delta f_{i-1} \begin{pmatrix} s \\ 1 \end{pmatrix} \right] hds \\ &= hf_{i-1} \int_0^1 ds + h(f_i - f_{i-1}) \int_0^1 sds \\ &= hf_{i-1} s \Big|_0^1 + h(f_i - f_{i-1}) \frac{s^2}{2} \Big|_0^1 \\ &= hf_{i-1} + h(f_i - f_{i-1}) \frac{1}{2} \\ I(p_1) &= \frac{h}{2}(f_{i-1} + f_i) \end{aligned} \tag{2.3}$$

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

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

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,

$$\begin{aligned} E(p_1) &\approx f[x_{i-1}, x_i, x_{i+1}] \int_{x_{i-1}}^{x_i} \left(x^2 - x(x_{i-1} + x_i) + x_{i-1}x_i\right) dx \\ &\approx \frac{1}{2!} f''(\xi) \frac{-h^3}{6} = O\left(h^3\right),\quad \xi \in (x_{i-1}, x_{i+1})\,. \end{aligned}$$

La expresión anterior significa que la diferencia entre el valor exacto de la integral y el valor aproximado es una constante por h^3 por $f''(\xi)$. De ahí que la fórmula de los trapecios sea 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:

$$I(p_1) = \sum_{i=1}^{m} \frac{h}{2}(f_{i-1} + f_i) = \frac{h}{2}\left[f_0 + 2f_1 + 2f_2 + ... + 2f_{m-1} + f_m\right]. \tag{2.4}$$

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

$$E_{TC} = -\frac{1}{12}h^3 f''(\xi_0) - \frac{1}{12}h^3 f''(\xi_1) - ... - \frac{1}{12}h^3 f''(\xi_{m-1}) \tag{2.5}$$

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

$$\begin{aligned} E_{TC} &= -\frac{h^2}{12}(b-a)\sum_{i=0}^{m-1} f''(\xi_i) = -\frac{h^2}{12}(b-a)f''(\xi) \\ &= O\left(h^2\right) \end{aligned} \tag{2.6}$$

con $\xi \in [a, b]$. Para obtener la expresión en (2.6) se aplica la siguiente generalización del teorema del valor medio.

Teorema 31 *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*

$$\sum_{i=1}^{n} f(x_i)g_i = f(\xi)\sum_{i=1}^{n} g_i \ , \ \xi \in [a,b] \tag{2.7}$$

Fórmula de Simpson en $[x_{2i-2}, x_{2i}]$, n=2

Aproximando f por una parábola de interpolación con nodos x_{2i-2}, x_{2i-1} y x_{2i} se obtiene:

$$\begin{aligned} I(p_2) &= \int_{x2_{i-2}}^{x_{2i}} p_2(x)\,dx \\ &= \int_{x2_{i-2}}^{x_{2i}} \{f(x_{2i-2}) + f[x_{2i-2}, x_{2i-1}](x - x_{2i-2}) + f[x_{2i-2}, x_{2i-1}, x_{2i}](x - x_{2i-2})(x - x_{2i-1})\}\,dx \end{aligned}$$

Si los nodos son equidistantes, entonces $h = x_i - x_{i-1}, \forall i$ y efectuando la integración se obtiene la regla básica de Simpson:

$$I(p_2) = \frac{h}{3}[f_{2i-2} + 4f_{2i-1} + f_{2i}] \tag{2.8}$$

El error de método en cada subintervalo $[x_{2i-2}, x_{2i}]$ estará dado por

$$\begin{aligned} E(p_2) &\approx \int_{x2_{i-2}}^{x_{2i}} f[x_{2i-2}, .., x_{2i+1}] \prod_{j=2i-2}^{2i} (x - x_j)\,dx \\ &= -\frac{h^5}{90} f^{(iv)}(\xi) = O\left(h^5\right), \ \xi \in (x_{2i-2}, x_{2i+1}) \end{aligned}$$

es decir el error de la regla básica es de orden 5.

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,

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

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

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

$$E_{SC} = \frac{(b-a)h^5}{(2h)90} f^{(iv)}(\xi) = \frac{h^4}{180}(b-a) f^{(iv)}(\xi). \tag{2.9}$$

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

$$I(f) - I(p_1) = -\frac{b-a}{12} h^2 f''(\xi) = O\left(h^2\right) \tag{2.10}$$
$$x_{i-1} < \xi < x_i$$

y para la fórmula de Simpson

$$I(f) - I(p_2) = -\frac{b-a}{180} h^4 f^{(iv)}(\xi) = O\left(h^4\right) \tag{2.11}$$
$$x_{2i-2} < \xi < x_{2i}.$$

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. A continuación se verá una forma de hacerlo.

2.2.3. 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:

$$E_h = I(f) - I_h(p) = ch^r f^{(s)}(\xi) = O(h^r) \tag{2.12}$$

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:

$$I(f) - I_{2h}(p_1) \approx (2h)^2 k + O\left(h^2\right) \tag{2.13}$$

$$I(f) - I_h(p_1) \approx (h)^2 k + O\left(h^2\right) \tag{2.14}$$

Restando (2.14) de (2.13) y despejando h^2k

$$h^2 k \approx \frac{I_h(p_1) - I_{2h}(p_1)}{3}. \tag{2.15}$$

Sustituyendo (2.15) en (2.14), 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 $[x_{i-2}, x_i]$:

$$E_h(p_1) = I(f) - I_h(p_1) \approx \frac{1}{3}\left[I_h(p_1) - I_{2h}(p_1)\right] \tag{2.16}$$

Observe que (2.16) 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

$$E_h(p) = I(f) - I_h(p) \approx \frac{1}{2^r - 1}\left[I_h(p) - I_{2h}(p)\right] \tag{2.17}$$

Las expresiones (2.16) y (2.17), 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.

2.2.4. Extrapolación de Richardson

El método de extrapolación de Richardson, desarrollado por Lewis Fry Richardson (1881-1953), permite construir a partir de una secuencia convergente otra secuencia 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 (2.17) 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:

$$I_0(p_1) = \frac{1}{3}\left[4I_h(p_1) - I_{2h}(p_1)\right] \tag{2.18}$$

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

$$I_0(p) = \frac{1}{2^r - 1}\left[2^r I_h(p) - I_{2h}(p)\right] \tag{2.19}$$

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.

2.2.5. 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

$$I_h(p_1) = (b-a)\left[\frac{1}{2}f(a) + \frac{1}{2}f(b)\right] \tag{2.20}$$

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

$$I_{\frac{h}{2}}(p_1) = \frac{b-a}{2}\left[\frac{1}{2}f(a) + \frac{1}{2}f(b) + f\left(a + \frac{b-a}{2}\right)\right] \tag{2.21}$$

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

$$T_1^0 = \frac{1}{2}T_0^0 + \frac{b-a}{2}f\left(a + \frac{b-a}{2}\right) \tag{2.22}$$

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

$$T_2^0 = \frac{b-a}{4}\left[\frac{1}{2}f(a) + \frac{1}{2}f(b) + f\left(a + \frac{b-a}{4}\right) + f\left(a + 2\frac{b-a}{4}\right) + f\left(a + 3\frac{b-a}{4}\right)\right]. \tag{2.23}$$

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 ,

$$T_2^0 = \frac{1}{2}T_1^0 + \frac{b-a}{4}\left[f\left(a+\frac{b-a}{4}\right) + f\left(a+3\frac{b-a}{4}\right)\right]. \tag{2.24}$$

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:

$$T_N^0 = \frac{1}{2}T_{N-1}^0 + h\sum_{i=1}^{2^{(N-1)}} f\left(a+(2i-1)h\right), N \geq 1 \tag{2.25}$$

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

$$T_N^1 = \frac{4T_N^0 - T_{N-1}^0}{3}, \quad N \geq 1. \tag{2.26}$$

Por ejemplo, para $N = 1$ obtenemos

$$\begin{aligned} T_1^1 &= \frac{4T_1^0 - T_0^0}{3} \\ &= \frac{h}{3}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] \end{aligned} \tag{2.27}$$

Como se observa (2.27) 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

$$T_N^2 = \frac{16T_N^1 - T_{N-1}^1}{15}, \text{ para } N \geq 2 \tag{2.28}$$

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

$$T_N^3 = \frac{64T_N^2 - T_{N-1}^2}{63}, \text{ para } N \geq 3 \tag{2.29}$$

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

$$T_N^j = \frac{4^j T_N^{j-1} - T_{N-1}^{j-1}}{2^{2j} - 1}, \text{ para } N \geq j \tag{2.30}$$

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 32

Poner $b-a \to h$
Calcular $\frac{h}{2}\left[f\left(a\right)+f\left(b\right)\right] \to T_0$
Para $n = 1, 2, ..$
$\frac{h}{2} \to n$
calcular $\frac{1}{2}T_{n-1} + h\sum_{i=1}^{2^{n-1}} f\left(a+\left(2i-1\right)h\right) \to T_n$
$n \to k$
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 33 *Calcular* $\int_1^5 \frac{1}{x}dx$ *por el algoritmo de Romberg.*

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

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

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

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

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

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

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. 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.

2. 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

$$P(a \leq X \leq b) = \int_a^b f(x)dx \tag{2.31}$$

   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

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{(2\sigma^2)}} \tag{2.32}$$

   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.

Capítulo 3

Ecuaciones diferenciales ordinarias

Introducción

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

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

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 la segunda forma, la derivada de mayor orden $y^{(n)}$ aparece expresada explícitamente en términos de $x, y, y', ..., y^{(n-1)}$. En la primera forma, $y^{(n)}$ está expresada implícitamente.

La resolución de la ecuación diferencial (3.1) consiste en la determinación de las funciones $y(x)$ que satisfacen esta ecuación para todos los valores de x en un determinado intervalo (a, b).

La solución general de una ecuación diferencial lineal no homogénea con coeficientes constantes tiene la forma

$$y(x) = \sum_{i=1}^{n} c_i y_i(x) + y_{part}(x),$$

donde $y_i(x)$, $i = 1, \ldots, n$, es el sistema fundamental de soluciones. Esta expresión representa una familia de funciones. Para la determinación de las constantes c_i de soluciones particulares es necesario considerar dos tipos de problemas:

1. problema de condiciones iniciales o problema de Cauchy:

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

2. problema de condiciones de frontera o de contorno:

$$y^{(n)} = f(x, y, y', ..., y^{(n-1)}), \quad x \in [a, b]$$

$$\sum_{k=o}^{n-1} (\alpha_{ik} y^{(k)}(a) + \beta_{ik} y^{(k)}(b)) = \gamma_i, \quad 1 \leq i \leq n.$$

En este último, los coeficientes α, β y γ están dados por las condiciones específicas del problema en los puntos $x = a$ y $x = b$ de la frontera.

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 :

- los métodos exactos resuelven sólo ecuaciones diferenciales con coeficientes constantes y algunas con coeficientes variables
- los métodos exactos no son aplicables cuando los coeficientes son empíricos
- no existen métodos exactos generales para la resolución de ecuaciones no lineales

Comenzaremos el estudio con los métodos numéricos para resolver el problema de valores iniciales.

3.1. Problema de Cauchy

Consideremos el problema de Cauchy en el intervalo [a,b] para los siguientes casos:

1. *a*) para una EDO de primer orden

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

b) para un sistema de EDOs de primer orden

$$\begin{cases} y_1' = f_1(x, y_1, \ldots, y_n),\ y_1(a) \\ y_2' = f_2(x, y_1, \ldots, y_n),\ \ y_2(a) \\ \qquad\qquad \cdots \\ y_n' = f_n(x, y_{1,\cdots}, y_n),\ \ y_n(a) \end{cases}$$

c) para una EDO de orden n

$$y^{(n)} = f(x, y, y', \ldots, y^{(n-1)}), \quad y(a), y'(a), \cdots, y^{(n-1)}(a)$$

Deduciremos métodos numéricos para resolver sólo el primer caso, ya que los otros dos se pueden resolver como extensión de éstos. Los tipos fundamentales de métodos numéricos son:

- los de paso simple, paso-por-paso o de pasos aislados, que se basan en el desarrollo en serie de Taylor de la solución $y(x + h)$ en términos de $y(x)$ y entre los cuales están el de Euler y los de Runge-Kutta.
- los de paso múltiple (*multistep*) o de pasos ligados, que siguen el camino de integrar la ecuación diferencial, que comprenden los de Adams y el esquema predictor-corrector.

Para la aplicación de estos métodos, se divide el intervalo de definición de la ecuación diferencial $[a, b]$, en N partes iguales de longitud $h = (b - a)/N$, que es el paso de integración, que determina una partición equidistante de dicho intervalo:

$$x_o = a,\ x_N = b, \quad x_i = x_o + ih, \quad (1 \leq i \leq N - 1),$$

y como solución numérica se obtiene un vector

$$y_h = (y(a), y_1, y_2, \ldots, y_N)^T$$

cuyas componentes son valores aproximados de la solución en los puntos x_i. En los métodos de paso simple, el valor aproximado y_{i+1} depende sólo del valor aproximado anterior y_i, y en los de paso múltiple, depende de $k + 1$ valores anteriores $y_i, y_{i-1,\ldots,}y_{i-k}$.

Lo más sencillo para tratar de resolver de forma aproximada el problema de valor inicial (3.3)

$$y' = f(x, y),\ y(a) \tag{3.4}$$

tomando $a = x_0$, es aplicar uno de los esquemas más simples vistos para la aproximación numérica de la primera derivada. Por ejemplo

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

de ahí que

$$y(x_0 + h) \approx y(x_0) + hf(x_0, y(x_0)) \tag{3.5}$$

La expresión 3.5 es lo que más adelante formalizaremos como método de Euler explícito. Si aproximamos la derivada con el modelo 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

$$y(x_0) \approx y(x_0 - h) + hf(x_0, y(x_0)). \tag{3.6}$$

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.

3.1.1. Integración por serie de Taylor

El desarrollo en serie de Taylor proporciona un método analítico aproximado para la resolución del problema de Cauchy de primer orden. Dada la ecuación diferencial $y' = f(x, y)$ y sea la función

f suficientemente diferenciable con respecto a x y a y. Si $y(x)$ es la solución exacta del problema de Cauchy (3.3),

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

$y(x)$ 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 :

$$y' = f \quad \prec \ {}^{x}_{y \longrightarrow x}$$

Así, para las primeras derivadas obtenemos:

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

Ejemplo 34 *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[1] es el más sencillo de los métodos de paso simple para resolver el problema de Cauchy de primer orden. 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

$$x_0, x_1, ..., x_{n-1}, x_n = X \tag{3.7}$$

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

$$\begin{aligned} y_1 - y_0 &= (x_1 - x_0)\, f\,(x_0, y_0) \\ y_2 - y_1 &= (x_2 - x_1)\, f\,(x_1, y_1) \\ &\vdots \\ y_n - y_{n-1} &= (x_n - x_{n-1})\, f\,(x_{n-1}, y_{n-1}) \end{aligned} \tag{3.8}$$

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 $a = x_o$,

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

[1]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:

$$y_{i+1} = y_i + h\ f(x_i, y_i),\ i = 0, 1, 2, \cdots \tag{3.9}$$

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 lo que nos interesa es el error que se comete al aproximar la solución exacta en el nodo x_i, $y(x_i)$ por la solución aproximada $y_h(x_i)$. En el cálculo de este error influye el que se comete en cada paso conocido como el **error de discretización local**, así como la estabilidad del algoritmo; aspectos 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 como 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

$$y_{i+1} = y_i + h\ \Phi(x_i, y_i, h_i),\ i = 0, 1, 2, \cdots \tag{3.10}$$

Si denotamos la solución exacta en el tiempo x_{i+1} como $y(x_{i+1})$ y asumimos se conoce $y(x_i)$, entonces

$$\begin{aligned} \frac{y(x_{i+1}) - y_{i+1}}{h} =& \frac{y(x_{i+1}) - (y_i + h\ \Phi(x_i, y_i, h_i))}{h} \\ =& \frac{y(x_{i+1}) - y(x_i)}{h} - \Phi(x_i, y(x_i), h_i) \\ =& L(x_i, h) \end{aligned} \tag{3.11}$$

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 (3.11) es una medida de cuán bien la solución exacta satisface el método de un solo paso dado.

Para el método de Euler donde $\Phi(x_i, y(x_i), h_i) = f(x_i, y(x_i))$ 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

$$\delta_{i+1}(x_i, y_h^i, h) = \Delta(x_i, y_h^i, h) - \Phi(x_i, y_i, h) \tag{3.12}$$

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

Definición 35 *Un método de un solo paso se dice consistente con el problema de valores iniciales o consistente de orden p si*

$$max_{x_i\in[a,b]}\ \|\delta_{i+1}\| = 0 \tag{3.13}$$

o

$$max_{x_i\in[a,b]}\ \|\delta_{i+1}\| = O(h^p) \tag{3.14}$$

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 el valor aproximado, por y_i , 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

$$y(x_{i+1}) - y_{i+1} = \{y(x_i) - y_i + h\ [f(x_i, y(x_i)) - f(x_i, y_i)]\} + \frac{h^2}{2}\ y''(\xi). \tag{3.15}$$

En (3.15) 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 (3.15) representa el error propagado (EP) y aplicando el teorema del valor medio para la variable y se obtiene

$$\begin{aligned} EP &= y(x_i) - y_i + h\ [\frac{\partial f}{\partial y}(x_i, \eta)(y(x_i) - y_i)] \\ &= (y(x_i) - y_i)\ (1 + h\ J_i), \end{aligned} \tag{3.16}$$

donde J_i denota la derivada parcial en (x_i, η), y $(1 + h\ J_i)$, el factor de propagación.

Sustituyendo (3.16) en (3.15), 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

$$z_1 - z_0 = (x_1 - x_0)\, f(x_0, z_0) \tag{3.17}$$

Necesitamos estimar $|z_1 - y_1|$. Restando (3.17) de la primera línea de (3.8)

$$\begin{aligned} y_1 - y_0 - z_1 + z_0 &= (x_1 - x_0)\, f(x_0, y_0) - (x_1 - x_0)\, f(x_0, z_0) \\ z_1 - y_1 &= -y_0 + z_0 + (x_1 - x_0)\,[f(x_0, z_0) - f(x_0, y_0)] \end{aligned}$$

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

$$z_1 - y_1 = (z_0 - y_0)\left\{1 + h\frac{\partial f}{\partial y}(x_0, \eta_0)\right\} \tag{3.18}$$

Si comparamos (3.18) con (3.16) obtenemos que la estabilidad del método de Euler dependerá de si el factor de propagación$1 + h\ J_i$ 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 desarrollaremos a continuación.

3.2. Los métodos de Runge-Kutta

La idea de los métodos de Runge-Kutta [2], ideados 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 $[x_i, x_i + h]$:

$$\begin{aligned} y_{i+1} &= y_i + \Delta y_i \\ &= y_i + \sum_{m=1}^{r} p_m k_m \end{aligned} \tag{3.19}$$

donde

$$k_m = h\ f(\xi_m, \eta_m)$$

$$\xi_m = x_i + \alpha_m h, \quad \alpha_1 = 0 \ \text{ por definición, y } 0 < \alpha_m \leq 1 \text{ para } m > 1$$

$$\eta_m = y_i + \beta_{m1}k_1 + \beta_{m2}k_2 + \cdots + \beta_{m,m-1}k_{m-1} = y_i + \sum_{j=1}^{m-1} \beta_{mj}k_j.$$

Los parámetros α, β y p 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 (3.19) coincida con el que se obtendría evaluando el desarrollo en serie de Taylor hasta el término de orden r :

$$y_{i+1} = y_i + \sum_{m=1}^{r} \frac{h^m}{m!} y_i^{(m)} \tag{3.20}$$

lo cual equivale a exigir que el incremento de Runge coincida con el incremento de Taylor:

$$\sum_{m=1}^{r} p_m k_m = \sum_{m=1}^{r} \frac{h^m}{m!} y_i^{(m)}. \tag{3.21}$$

Comparando las expresiones (3.19) y (3.20), se observa que en la primera, 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.

[2] 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 Runge-Kutta de segundo orden

Siendo $r = 2$, el incremento estará definido por

$$\Delta y_i = p_{21}k_1 + p_{22}k_2$$

donde

$$\begin{aligned} k_1 &= h\ f(\xi_1, \eta_1) = h\ f(x_i, y_i) \\ k_2 &= h\ f(\xi_2, \eta_2) = h\ f(x_i + \alpha_2 h, y_i + \beta_{21}k_1). \end{aligned}$$

Para simplificar la notación, usemos p_1, p_2, α y β en lugar de p_{21}, p_{22}, α_2 y β_{21}. Los coeficientes desconocidos se determinan de manera que

$$y_{i+1} = y_i + p_1k_1 + p_2k_2 \tag{3.22}$$

coincida con el desarrollo de Taylor en el punto x_i hasta el término de segundo orden:

$$y_{i+1} = y_i + hy_i' + \frac{h^2}{2}y_i''. \tag{3.23}$$

Sustituyendo $y' = f(x, y),\ \ y'' = \frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y}$ en (3.23), obtenemos

$$y_{i+1} = y_i + h\ f(x_i, y_i) + \frac{h^2}{2}(\frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y})(x_i, y_i) \tag{3.24}$$

Por otra parte, usando el desarrollo en serie de Taylor para la función de dos variables $k_2(\alpha, \beta)$ obtenemos

$$\begin{aligned} k_2(x_i + \alpha h, y_i + \beta k_1) &= h\ f(x_i + \alpha h, y_i + \beta k_1) \\ &= h[f(x_i, y_i) + (\alpha h\frac{\partial}{\partial x} + \beta k_1\frac{\partial}{\partial y})\ f(x_i, y_i) \\ &+ \frac{1}{2!}(\alpha h\frac{\partial}{\partial x} + \beta k_1\frac{\partial}{\partial y})^2\ f(x_i, y_i) + \cdots \end{aligned}$$

Sustituyendo k_1 por $hf(x_i, y_i)$ y k_2 por la expresión anterior en (3.22), obtenemos después de agrupar en potencias de h :

$$y_{i+1} = y_i + h(p_1 + p_2)\ f(x_i, y_i) + h^2[p_2(\alpha\frac{\partial f}{\partial x} + \beta f\frac{\partial f}{\partial y})(x_i, y_i)] + \Theta(h^3). \tag{3.25}$$

Igualando (3.24) y (3.25) término a término, obtenemos para los coeficientes de h

$$(p_1 + p_2)f(x_i, y_i) = f(x_i, y_i) \Longrightarrow p_1 + p_2 = 1,$$

y para los coeficientes de h^2

$$p_2(\alpha\frac{\partial f}{\partial x} + \beta f\frac{\partial f}{\partial y})(x_i, y_i) = \frac{1}{2}(\frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y})(x_i, y_i) \Longrightarrow p_2\alpha = \frac{1}{2},\ \ p_2\beta = \frac{1}{2},$$

es decir, tenemos 3 ecuaciones para la determinación de las 4 incógnitas p_1, p_2, α, β, luego existe un grado de libertad.

Tomando $p_2 = c$, constante arbitraria, tal que $0 < \alpha \leq 1$, según las hipótesis de los métodos de Runge-Kutta tenemos que $\alpha = \frac{1}{2c} \leq 1$, o sea, $c \geq \frac{1}{2}$. Entonces queda

$$p_2 = c,\ p_1 = 1 - c,\ \alpha = \beta = \frac{1}{2c},\ \text{con}\ \ c \geq \frac{1}{2}. \tag{3.26}$$

Casos particulares

$c = \frac{3}{4} \implies p_2 = \frac{3}{4}$, $p_1 = \frac{1}{4}$, $\alpha = \beta = 2/3$:

$$y_{i+1} = y_i + \frac{1}{4}(k_1 + 3k_2),$$
$$k_1 = hf(x_i, y_i),\ k_2 = h\ f(x_i + \frac{2}{3}h, y_i + \frac{2}{3}k_1)$$

$c = \frac{1}{2} \implies p_2 = \frac{1}{2}$, $p_1 = \frac{1}{2}$, $\alpha = \beta = 1$:

$$y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2),$$
$$k_1 = hf(x_i, y_i),\ k_2 = h\ f(x_i + h, y_i + k_1)$$

$c = 1 \implies p_2 = 1$, $p_1 = 0$, $\alpha = \beta = \frac{1}{2}$:

$$y_{i+1} = y_i + k_2,$$
$$k_1 = hf(x_i, y_i),\ k_2 = h\ f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1)$$

Para otros valores admisibles de c se obtienen otras fórmulas de Runge-Kutta de 2do. orden, todas con error de método $\Theta(h^3)$.

3.3. Fórmulas de orden superior

Análogamente a como se hizo para deducir las fórmulas de Runge-Kutta de orden 2, pueden deducirse 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

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{3.27}$$

donde

$$\begin{aligned} k_1 &= h\ f(x_i, y) \\ k_2 &= h\ f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1) \\ k_3 &= h\ f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2) \\ k_4 &= h\ f(x_i + h, y_i + \frac{1}{2}k_3) \end{aligned} \tag{3.28}$$

con error de método $\Theta(h^5)$, que se obtiene a partir de un sistema de 11 condiciones para la determinación de las 13 incógnitas $p_{41}, p_{42}, p_{43}, p_{44}$; $\alpha_2, \alpha_3, \alpha_4$; $\beta_{21}, \beta_{31}, \beta_{32}, \beta_{41}, \beta_{42}, \beta_{43}$.

Siguiendo el patrón de Butcher para las fórmulas de orden 4:

$$\begin{bmatrix} \alpha_1 & 0 & 0 & 0 & 0 \\ \alpha_2 & \beta_{21} & 0 & 0 & 0 \\ \alpha_3 & \beta_{31} & \beta_{32} & 0 & 0 \\ \alpha_4 & \beta_{41} & \beta_{42} & \beta_{43} & 0 \\ - & p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ - & p \end{bmatrix},$$

donde α es un vector columna de 4 componentes, β una matriz 4×4 estrictamente triangular inferior, y p, un vector fila de 4 componentes, la fórmula anterior se representaría por:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ - & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{bmatrix}. \tag{3.29}$$

3.3.1. Esquema de cálculo

La aplicación de una fórmula de Runge-Kutta de orden r requiere r evaluaciones de la función $f(x, y)$ en cada paso para obtener $k_1, k_2, \cdots, k_r$. Para organizar los cálculos será conveniente 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	—	$\Delta\mathrm{y}_{i-1}$
-	$\xi_1 = \mathrm{x}_i$	$\eta_1 = \mathrm{y}_i$	$\mathrm{f}(\xi_1, \eta_1)$	k_1
-	$\xi_2 = \mathrm{x}_i + \alpha_2 h$	$\eta_2 = \mathrm{y}_i + \beta_{21}\mathrm{k}_1$	$\mathrm{f}(\xi_2, \eta_2)$	k_2
-	$\vdots$	$\vdots$	$\vdots$	$\vdots$
-	$\xi_r = \mathrm{x}_i + \alpha_r h$	$\eta_r = \mathrm{y}_i + \beta_{r1}\mathrm{k}_1 + \cdots + \beta_{r,r-1}\mathrm{k}_{r-1}$	$\mathrm{f}(\xi_r, \eta_r)$	k_r
i+1	$\mathrm{x}_{i+1} = \mathrm{x}_i + h$	$\mathrm{y}_{i+1} = \mathrm{y}_i + \Delta\mathrm{y}_i$	—	$\Delta\mathrm{y}_i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	$\mathrm{x}_n = \mathrm{x}_{n-1} + h$	$\mathrm{y}_n = \mathrm{y}_{n-1} + \Delta\mathrm{y}_{n-1}$	—	$\Delta\mathrm{y}_{n-1}$

Ejemplo

Resolver el problema de Cauchy

$$y' = y\sin(x) + \cos(x) + 1, y(1) = 1,\ x \in [1, 2]$$

usando la fórmula de Runge-Kutta de orden 2:

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{2}(k_1 + k_2), \\ k_1 &= hf(x_i, y_i),\ k_2 = h\ f(x_i + h, y_i + k_1) \end{aligned}$$

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
⋮	⋮	⋮	⋮	⋮	⋮	⋮
10	2.0	⋯	—	—	—	Δy_9

3.3.2. Algoritmo de Runge-Kutta con paso fijo

Problema de Cauchy: $y' = f(x,y),\ \ x \in [a,b],\ \ y(a)$
Parámetros fijos : α, β, p de un método de orden r
Subrutina auxiliar : $f(x,y)$
Paso fijo de integración: $h = (b-a)/c$, donde c es la cantidad de pasos
Paso 1: leer a, b, c 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 $i = 1$ hasta c, repetir:
| 5.1) poner $x \curvearrowright \xi$, $y \curvearrowright \eta$, calcular $h\ f(\xi,\eta)$, y guardarlo en k_1
| ⌈ para $m = 2$ hasta r,
| | calcular $x + \alpha_m h \curvearrowright \xi$
| | calcular $\sum_{j=1}^{m-1} \beta_{mj} k_j \curvearrowright z,\ \ y + z \curvearrowright \eta$
| | calcular $\ h\ f(\xi,\eta)$ 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

3.4. Estimación del error

Anteriormente se mencionó que una fórmula de Runge-Kutta de orden r requiere r evaluaciones de la función f por paso, con lo cual se evita la evaluación de las derivadas, de segundo orden, hasta la de orden r, cada paso. Para la estimación del error local, 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.

En la práctica, la estimación del error puede obtenerse mediante el procedimiento del doble cómputo, que fue usado para estimar el error en la integración numérica, o usando simultáneamente dos fórmulas de paso simple de distinto orden, (por ejemplo, el algoritmo RKF45).

3.4.1. 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 $\Theta(h^{r+1}) = \frac{h^{r+1}}{(r+1)!}y^{(r+1)}(\xi)$ en cada paso, y el error global en el punto $\overline{x} = x_i + mh$ se expresa de la forma

$$E_g(\overline{x}) = y(\overline{x}) - y_h(\overline{x}) = C(\overline{x})\ h^r + \Theta(h^{r+1}). \tag{3.30}$$

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 (3.30) 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 (3.30) con $m = 1$, y paso h, es decir, efectuando un solo paso:

$$E_g = y(x_{i+1}) - y_h(x_{i+1}) = C(x_{i+1})h^r + \Theta(h^{r+1}), \tag{3.31}$$

y con paso de amplitud $h/2$, efectuando dos cálculos

$$E_g = y(x_{i+1}) - y_{h/2}(x_{i+1}) = C(x_{i+1})(h/2)^r + \Theta((h/2)^{r+1}). \tag{3.32}$$

Restando (3.32) de (3.31),

$$y_{h/2}(x_{i+1}) - y_h(x_{i+1}) \approx C(x_{i+1})h^r(1 - \frac{1}{2^r}), \tag{3.33}$$

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 (3.31), obtener

$$E_g \approx \frac{2^r(y_{h/2}(x_{i+1}) - y_h(x_{i+1})}{2^r - 1}, \tag{3.34}$$

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.

3.4.2. 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 $\Theta(h^4)$ y $\Theta(h^5)$ respectivamente:

$$y_{i+1}^{(5)} = y_i^{(5)} + \sum_{j=1}^{5} p_{5j} k_j$$
$$y_{i+1}^{(6)} = y_i^{(6)} + \sum_{j=1}^{6} p_{6j} k_j,$$

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 α_m y los β_{mj} comunes son iguales, luego los k_j coinciden para $1 \leq j \leq 5$, mientras que los p_{5j} y p_{6j} difieren. La tabla de parámetros según el patrón de Butcher tendrá la forma:

$$\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha_2 & \beta_{21} & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & 0 & 0 & 0 \\
\alpha_5 & \beta_{51} & \beta_{52} & \beta_{53} & \beta_{54} & 0 & 0 \\
\alpha_6 & \beta_{61} & \beta_{62} & \beta_{63} & \beta_{64} & \beta_{65} & 0 \\
- & p_{51} & p_{52} & p_{53} & p_{54} & p_{55} & 0 \\
- & p_{61} & p_{62} & p_{63} & p_{64} & p_{65} & p_{66}
\end{array}$$

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 $h/2$).

3.4.3. Aplicación al cambio de paso

Después de calcular $x_i + h \longrightarrow \overline{x}$, $y_i + \sum_{j=1}^{r} p_{rj} k_j \longrightarrow u$, 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 $[a, b]$, (intervalo de definición del problema de Cauchy), la solución $y(x)$ presenta variaciones grandes, cosa que 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 $\varepsilon_1 = \varepsilon_2/2^5 = \varepsilon_2/32$.

3.4.4. Incorporación del cambio de paso al algoritmo de Runge-Kutta

Sustituir el paso 5 por:

Paso 5: ⌈ mientras $x < b$), repetir:

| 5.1) poner $x \curvearrowright \xi$, $y \curvearrowright \eta$, calcular $h\ f(\xi, \eta)$ y guardarlo en k_1

| ⌈ para $m = 2$ hasta r,

| | calcular $x + \alpha_m h \curvearrowright \xi$

| | calcular $\sum_{j=1}^{r-1} \beta_{rj} k_j \curvearrowright z$, $y + z \curvearrowright \eta$

| | calcular $h\ f(\xi, \eta)$ y guardarlo en k_m

| ⌊ fin

| 5.2) resultados del nuevo paso:

| calcular $x + h$ y guardarlo en $\overline{x}$ y x_i

| calcular $\sum_{j=1}^{r} p_{rj} k_j \curvearrowright z$, $y + z$, y guardarlo en y y en y_i

| 5.3) poner $y \curvearrowright u$, y 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.

3.4.5. Propiedades de los métodos de Runge-Kutta

El estudio realizado de los métodos de Runge-Kutta, representativos de los de paso simple, permite resumir sus propiedades, que después compararemos con las de los métodos de paso múltiple. Estas son:

1- Se autoinician, es decir, basta conocer la condición inicial para poder aplicarlos

2- En cada paso se realizan r evaluaciones de la parte derecha $f = y'$, con lo cual superan al 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

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.

Capítulo 4

Aproximación de funciones por mínimos cuadrados

Introducción

La teoría de aproximación como ya se conoce involucra dos tipos de problemas: el primero que ya hemos estudiado donde la aproximación de funciones se realiza por interpolación ya que se considera que la precisión de los datos es suficiente como para que se puedan considerar exactos. Sin embargo a pesar de tener datos exactos se demostró que no es de utilidad considerar un solo polinomio que interpole todos los datos cuando se tienen muchos datos; en este caso es conveniente la interpolación por tramos. Ahora bien si además de contar con muchos datos estos se consideran afectados de error ó 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 aproximada 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 se puede proponer aproximar por una función trigonométrica, y así según el caso.

Estos ejemplos nos permiten recordar los tres componentes fundamentales que se deben considerar para el cálculo de una aproximación, que contempla conocer:

1. si la función f que debe ser aproximada es continua o discreta
2. el conjunto Φ donde se seleccionan las funciones de aproximación F
3. la función error $f - F$ y la norma de la misma, que constituye la base para la selección de la mejor función de aproximación

Problema general de aproximación

El problema de aproximar una función se puede considerar en un contexto bien general; dado un espacio lineal normado L y un subespacio Φ del espacio lineal normado 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

$$\left\| f - \widehat{F} \right\| = \min_{F \in \Phi} \|f - F\| . \tag{4.1}$$

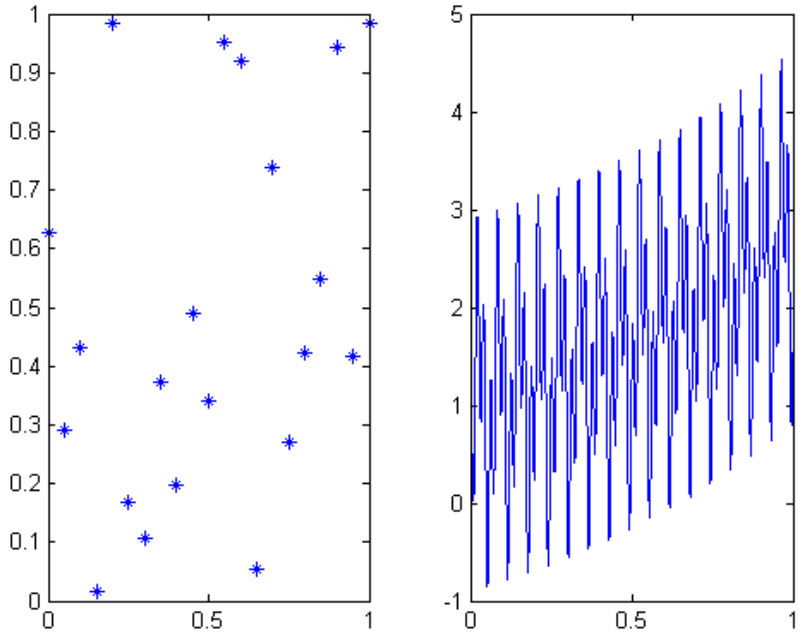

Figura 4.1:

Si la distancia entre las dos funciones (residual) se mide según la norma Euclideana, entonces estamos ante un problema de mínimos cuadrados (tomar otras normas define otros tipos de aproximación que no trataremos aquí, pero se pueden consultar en [], []). El problema de aproximación mínimo cuadrática se puede encontrar relacionado con diferentes aplicaciones como:

- Resolver $Ax = b$, $A_{m\times n}$, con $m > n$ ó $m < n$ que implica $min_x \|Ax - b\|_2$, (vista en el primer semestre)
- ajuste de curvas
- modelación estadística de datos con ruido
- modelación geodésica
- problema de optimización sin restricciones

Si se define el funcional $\varphi(F) = \|f - F\|$, $\varphi : \Phi \to \mathbb{R}$, entonces estaríamos ante un problema de optimización formulado de forma general abstracta y para asegurar la existencia y unicidad del valor extremo existen algunos resultados teóricos que aparecen en el apéndice.

Observación 36

- *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*

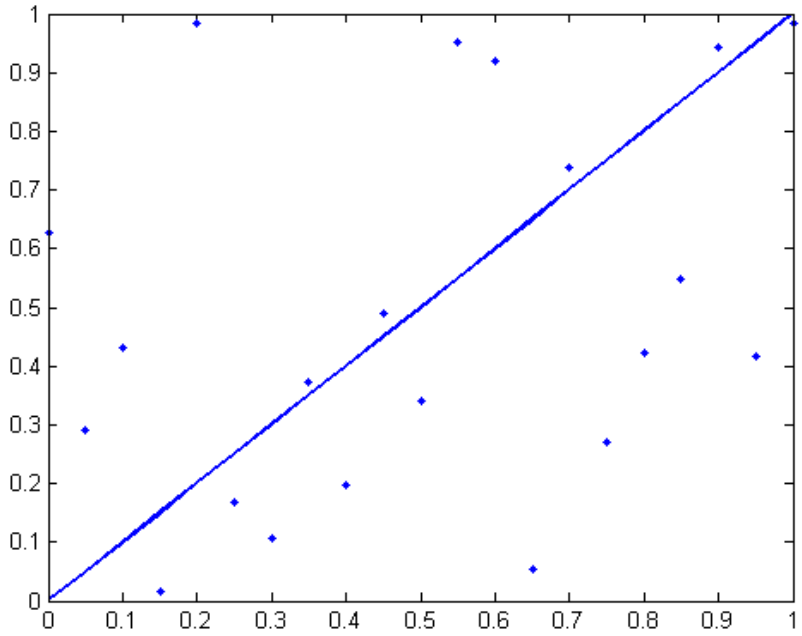

Figura 4.2:

- *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* $x, y \in B$*,* $\gamma > 0$*)*

Nosotros comenzaremos considerando el problema de mínimos cuadradros desde el punto de vista del ajuste de curvas.

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

$$\Phi = \left\{F(x,c) : \mathbb{R} \to \mathbb{C}, c \in \Re^{n+1}\right\}$$

Definición 37 *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*

$$r_{min} = \left\|f - \widehat{F}(x,c^*)\right\|_2 = min_{c\in\Re^{n+1}} \|f - F(x,c)\|_2$$

En el caso de la norma discreta (ver al final del documento expresiones de las normas), se tiene

$$\left\|f - \widehat{F}(x,c^*)\right\|_2 = min_{c\in\Re^{n+1}} \left[\sum_{i=0}^{N}(f(x_i) - F(x_i, c_0, c_1, ...c_n))^2\right]^{\frac{1}{2}}$$

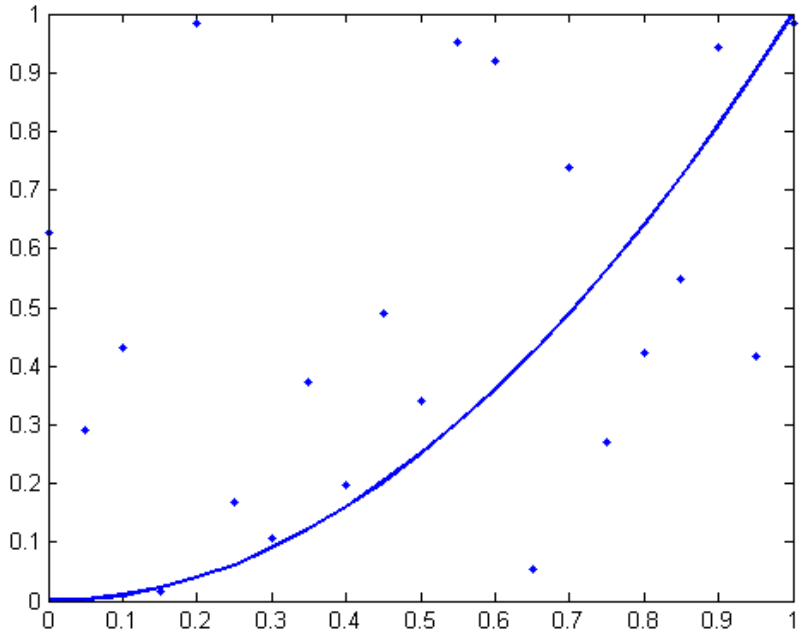

Figura 4.3:

El tratamiento del problema y las vías de solución están ahora relacionados con el hecho de que

- 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 $F(x, c), c \in \Re^{n+1}$ 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

$$\Phi = \left\{F(x, c) : \mathbb{R} \to \mathbb{C}, c \in \Re^{n+1}\right\}$$

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, es decir, cuando $F(x)$ depende linealmente de los coeficientes desconocidos c_j, entonces estamos ante un problema de aproximación mínimo cuadrática lineal, es decir el subespacio Φ es de dimensión finita y

está generado por $\{\varphi_k\}_{k=0}^{n}$. Entonces el problema a resolver es

$$\left\| f - \sum_{j=o}^{n} c_j^* \varphi_j(x) \right\|_2^2 = \min_{c \in \Re^{n+1}} \left\| f - \sum_{j=o}^{n} c_j \varphi_j(x) \right\|_2^2 \tag{4.2}$$

Se considera que la norma fue inducida por un producto escalar, $\|f\| = \sqrt{\langle f, f \rangle}$ y como producto escalar se tiene:

- producto escalar continuo $\langle f, g \rangle = \int_a^b f(x)\overline{g(x)}dx$
- producto escalar discreto $\langle f, g \rangle = \sum_{i=1}^{n} f(x_i)\overline{g(x_i)}$

Entonces el problema (4.2) se reduce a

$$\min_{c \in \Re^{n+1}} \langle f - F(x,c), f - F(x,c) \rangle$$

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

$$ming(c) \Leftrightarrow min(g(c))^2$$

$$\left\| f - \widehat{F}(x,c^*) \right\|_2 = \min_{c \in \Re^{n+1}} \|f - F(x,c)\|_2 \tag{4.3}$$

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 $\omega(x_i)$ respectivamente, que aparecen en las expresiones anteriores, afecta a $\widehat{F}$. En el caso *continuo,* con una elección adecuada se puede forzar a que $\widehat{F}$ concuerde mejor con 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 del valor f_i (en la terminología estadística, se dice que (x_i, f_i) es un punto mentiroso o *outlier* en este caso). 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 38 *Sean las funciones $\varphi_o, \varphi_1, ..., \varphi_n$ l.i. 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:

$$< f - \widehat{F}, \varphi_k >= 0, \quad 0 \leq k \leq n.$$

y viceversa.

Demostración:
Teniendo en cuenta las propiedades del producto escalar y la forma de $\widehat{F}$,

$$\begin{aligned} &< f - \widehat{F}, \varphi_k >= 0 \Leftrightarrow < \widehat{F}, \varphi_k >=< f, \varphi_k > \\ &0 \leq k \leq n \end{aligned}$$

sustituyendo $\widehat{F}$

$$< \sum_{j=o}^{n} c_j^* \varphi_j, \varphi_k >=< f, \varphi_k >$$

y teniendo en cuenta las propiedades asociativa y distributiva del producto escalar, se llega a que

$$\sum_{j=o}^{n} c_j^* < \varphi_j, \varphi_k >=< f, \varphi_k > , \ 0 \leq k \leq n, \tag{4.4}$$

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

$$Bc^* = h$$

$$B = \begin{bmatrix} < \varphi_o, \varphi_o > & < \varphi_1, \varphi_o > & \cdots & < \varphi_n, \varphi_o > \\ < \varphi_o, \varphi_1 > & < \varphi_1, \varphi_1 > & \cdots & < \varphi_n, \varphi_1 > \\ \cdots & \cdots & \cdots & \cdots \\ < \varphi_o, \varphi_n > & < \varphi_1, \varphi_n > & \cdots & < \varphi_n, \varphi_n > \end{bmatrix} \tag{4.5}$$

$$h = \begin{bmatrix} < f, \varphi_o > \\ < f, \varphi_1 > \\ \cdots \\ < f, \varphi_n > \end{bmatrix}$$

$c = (c_o^*, c_1^*, ..., c_n^*)^T$ y se conoce como sistema de las ecuaciones normales (SEN) o de Gauss. La denominación de normales proviene del hecho que

$$< f - \widehat{F}, \phi_k >= 0 \text{ equivale a que } f - \widehat{F} \perp \Phi \tag{4.6}$$

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. Nótese que debido a la conmutatividad del producto escalar, la matriz B es simétrica. (hacer gráfico)

La matriz B simétrica de los productos escalares del SEN (4.5) es una matriz de Gram, la cual se puede demostrar que es definida positiva siempre que las funciones φ_j sean linealmente independientes.

Definición 39 *Una matriz A es definida positiva (semidefinida positiva), $A \succ 0 (\succeq 0)$ si y sólo sí $x^T Ax > 0 (x^T Ax \geq 0)$ para toda $x \neq 0 (x \in \Re^n)$*

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 (4.6), 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.
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,

$$\left\langle \sum_{k=0}^{n} c_k\varphi_k(x) - f(x), \sum_{k=0}^{n} c_k\varphi_k(x) - f(x) \right\rangle = \sum_{k=0}^{n} c_k \sum_{j=0}^{n} c_j \left\langle \varphi_k(x), \varphi_j(x) \right\rangle \tag{4.7}$$

$$- 2\sum_{0}^{n} c_k Re(\langle f, \varphi_k(x) \rangle) + \langle f, f \rangle \tag{4.8}$$

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

$$\min_{c\in R^n} \sum_{k=0}^{n} c_k \sum_{j=0}^{n} c_j \left\langle \varphi_k(x), \varphi_j(x) \right\rangle - 2\sum_{0}^{n} c_k \left\langle f, \varphi_k(x) \right\rangle \tag{4.9}$$

Veamos cuáles son las condiciones de optimalidad para el problema

$$\min_{c\in\Re^n} g(c) \tag{4.10}$$

Definición 40

- *c^* es un mínimo global de (4.10) si $\forall\ c \in \Re^n$; $g(c) \geq g(c^*)$*
- *Si existe una vecindad V_{c^*} de c^* tal que $\forall\ c \in V_{c^*} \cap \Re^n$; $g(c) \geq g(c^*)$, entonces c^* es un mínimo local de (4.10)*

Definición 41 *La función $g(x)$ es convexa si $\forall x_1, x_2 \in \Re^n$, $\alpha \in [0, 1]$ 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 sólo sí $\nabla^2 g(c) \succ 0$.
La condición necesaria de mínimo local es como sigue:

Teorema 42 *Si c^* es mínimo local de (4.10) 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.*

Condiciones suficientes

Teorema 43 *Si $g \in C^2$, $\nabla g(c^*) = 0$ y $\nabla^2 g(c^*)$ es definida positiva entonces c^* es un mínimo local de (4.10).*

Si g es convexa, entonces la condición $\nabla g(c^*) = 0$ es condición suficiente para la existencia del mínimo global. Retomando nuestro problema (4.9). Nuestra función $g(c) \in C^\infty$

$$g(c) = \sum_{k=0}^{n} c_k \sum_{j=0}^{n} c_j \left\langle \varphi_k(x), \varphi_j(x) \right\rangle - 2\sum_{0}^{n} c_k \left\langle f, \varphi_k(x) \right\rangle \tag{4.11}$$

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 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 $F \epsilon P_n = \Phi$, entonces

$$F(x) = c_o + c_1 x + c_2 x^2 + \cdots + c_n x^n = \sum_{j=o}^{n} c_j x^j,$$

y se dispone de la función f que se quiere aproximar en la forma de una tabla de $N+1$ pares ($N >> n$, N grande): $\begin{matrix} \mathrm{x} & \mathrm{x}_o & \mathrm{x}_1 & \cdots & \mathrm{x}_N \\ \mathrm{f(x)} & \mathrm{f}_o & \mathrm{f}_1 & \cdots & \mathrm{f}_N \end{matrix}$ En este caso, se pueden tomar como funciones linealmente independientes las potencias de x:

$$\{\varphi_j(x)\}_{j=o}^{n} = \left\{x^j\right\}_{j=o}^{n}.$$

El producto escalar con función de peso $\omega(x) = 1$ está definido por

$$< \varphi_j, \varphi_k >= \sum_{i=o}^{N} \varphi_j(x_i)\varphi_k(x_i) = \sum_{i=o}^{N} x_i^j x_i^k = \sum_{i=o}^{N} x_i^{j+k}$$

y

$$< f, \varphi_k >= \sum_{i=o}^{N} f(x_i)\varphi_k(x_i) = \sum_{i=o}^{N} f_i x_i^k$$

obteniéndose, por ejemplo,

si $j = k = 0$, $< \varphi_o, \varphi_o >= \sum x_i^{o+o} = \sum 1 = N+1$

si $j = k = 1$ $< \varphi_1, \varphi_1 >= \sum x_i^{1+1} = \sum x_i^2$

si $j = 0, k = 1$ $< \varphi_o, \varphi_1 >= \sum x_i^{o+1} = \sum x_i$, etc.

El sistema (4.1.1) de las ecuaciones normales tendrá la forma

$$\begin{bmatrix} N+1 & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{n+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{n+2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \cdots & \sum x_i^{2n} \end{bmatrix} \begin{bmatrix} c_o^* \\ c_1^* \\ c_2^* \\ \vdots \\ c_n^* \end{bmatrix} \tag{4.12}$$

$$= \begin{bmatrix} \sum f_i \\ \sum x_i f_i \\ \sum x_i^2 f_i \\ \vdots \\ \sum x_i^n f_i \end{bmatrix}, \tag{4.13}$$

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 éstos 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:

$$X = \begin{bmatrix} 1 & x_o & x_o^2 & \cdots & x_o^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_N & x_N^2 & \cdots & x_N^n \end{bmatrix}, \qquad f = \begin{bmatrix} f_o \\ f_1 \\ \cdots \\ f_N \end{bmatrix}.$$

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

$$x = (x_o, x_1, \ldots, x_N)^T.$$

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 $\{\varphi_j\}$ ortogonales
- no usar las ecuaciones normales

Sobre esto volveremos después. **Aproximación mínimo cuadrática polinomial en la Estadística** Es atinado comentar que 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 y no es más que la aproximación mediante una recta, $\widehat{F}(x) = c_o^* + c_1^* x$ de un conjunto de datos

$$X = \begin{bmatrix} 1 & x_o \\ 1 & x_1 \\ \cdots & \cdots \\ 1 & x_N \end{bmatrix}, \qquad f = \begin{bmatrix} f_o \\ f_1 \\ \cdots \\ f_N \end{bmatrix}.$$

Aplicando la teoría vista más arriba se obtiene que el sistema de las ecuaciones normales tiene la forma

$$\begin{bmatrix} N+1 & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} c_o^* \\ c_1^* \end{bmatrix} = \begin{bmatrix} \sum f_i \\ \sum x_i f_i \end{bmatrix}$$

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 $F(x) = c_o e^{c_1 x}$. La función aproximante puede linealizarse aplicando logaritmos. Aplicando logaritmos se convierte en una recta:

$$\ln F(x) = \ln c_o + c_1 x,$$

o sea,

$$G(x) = c_o' + c_1 x,$$

donde $G(x) = \ln F(x)$ y $c_o' = \ln c_o$. Con la transformación logarítmica, se ha convertido la función aproximante F que depende en forma no lineal de c_o y c_1, en la función aproximante G, que depende de c_o' y c_1 linealmente. El problema de aproximación se convierte entonces en hallar $\widehat{G}$ tal que

$$\left\| \ln f - \widehat{G} \right\|_2^2 = \min_G \|\ln f - G\|_2^2 = \textstyle\sum_{i=o}^{N} [\ln f_i - (c_o' + c_1 x_i)]^2$$

Está claro que los coeficientes $c_o^* = \exp(c_o'^*)$ 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 $\{\varphi_j(x)\}_{j=o}^{1} = \{1, x\}$ y los datos representados por

$$X = \begin{bmatrix} 1 & x_o \\ 1 & x_1 \\ \cdots & \cdots \\ 1 & x_N \end{bmatrix}, \qquad f = \begin{bmatrix} \ln f_o \\ \ln f_1 \\ \cdots \\ \ln f_N \end{bmatrix},$$

el SEN será:

$$\begin{bmatrix} N+1 & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} c_o'^* \\ c_1^* \end{bmatrix} = \begin{bmatrix} \sum \ln f_i \\ \sum x_i \ln f_i \end{bmatrix},$$

cuya solución $(c_o'^*, c_1^*)$ permite determinar finalmente $c_o^* = e^{c_o'^*}$, y definir la función de aproximación $\widehat{F}(x) = c_o^* e^{c_1^* x}$.

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:

obs	x_1	x_2	$\cdots$	x_p	f
0	x_{o1}	x_{o2}	$\cdots$	x_{op}	f_o
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

$$F(x_1, x_2, \ldots, x_p) = c_o + c_1x_1 + c_2x_2 + \cdots + c_px_p,$$

y supuesto que los vectores $x_1, x_2, \ldots, x_p$ que definen las variables son linealmente independientes, puede considerarse el conjunto de funciones $\{\varphi_j(x)\}_{j=o}^{p} = \{1, x_1, x_2, \ldots, x_p\}$ y los datos representados por:

$X = \begin{bmatrix} 1 & x_{o1} & x_{o2} & \cdots & x_{op} \\ 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Np} \end{bmatrix}$, $f = \begin{bmatrix} f_o \\ f_1 \\ \vdots \\ f_N \end{bmatrix}$, con lo cual, el SEN tendrá la forma:

$$\begin{bmatrix} N+1 & \sum x_{i1} & \sum x_{i2} & \cdots & \sum x_{ip} \\ \sum x_{i1} & \sum x_{i1}^2 & \sum x_{i1}x_{i2} & \cdots & \sum x_{i1}x_{ip} \\ \sum x_{i2} & \sum x_{x2}x_{i1} & \sum x_{i2}^2 & \cdots & \sum x_{i2}x_{ip} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum x_{ip} & \sum x_{ip}x_{i1} & \sum x_{ip}x_{i2} & \cdots & \sum x_{ip}^2 \end{bmatrix} \begin{bmatrix} c_o^* \\ c_1^* \\ c_2^* \\ \vdots \\ c_p^* \end{bmatrix} = \begin{bmatrix} \sum f_i \\ \sum x_{i1}f_i \\ \sum x_{i2}f_i \\ \vdots \\ \sum x_{ip}f_i \end{bmatrix}.$$

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 $b = f \,\epsilon\, \Re^n$ 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. Esto ya se vio en la asignatura MNI. En este contexto se demuestra el siguiente resultado

Teorema 44 *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

$$\ln F = \ln \alpha + \beta \ln x + \gamma \ln y - \delta \ln z,$$

βo 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

Definamos la función S como el cuadrado del error de la aproximación mínimo cuadrática:

$$S = \|f - F\|_2^2 = \sum_{i=o}^N [f(x_i) - F(x_i; c_o, c_1, \ldots, c_n)]^2, \tag{4.14}$$

$$S = S(c_o, c_1, \ldots, c_n).$$

que es una función continua, positiva y diferenciable de los parámetros $c_o, c_1, \ldots, c_n$ por lo menos hasta de segundo orden, $S : \mathbb{R}^n \to \mathbb{R}$. Luego el problema formulado como sigue: encontrar $c^* \in \Re^n$ tal que

$$S(c^*) = \underset{c \in \Re^n}{min} S(c)$$

es un problema de optimización sin restricciones. Entonces se aplican las condiciones de optimalidad vistas anteriormente

$$\nabla S(c) = \left(\frac{\partial S}{\partial c_o}, \frac{\partial S}{\partial c_1}, \ldots, \frac{\partial S}{\partial c_n} \right)^T = \overrightarrow{0_{\Re^{n+1}}}$$

Teniendo en cuenta (**??**), y derivando con respecto a los c_j, se obtiene

$$\begin{aligned} \frac{\partial S}{\partial c_j} &= \sum_{i=o}^N \left\{ \frac{\partial}{\partial c_j} \left([f(x_i) - F(x_i; c_o, \ldots, c_n)]^2 \right) \right\} \\ &= -2 \sum_{i=o}^N \left\{ [f(x_i) - F(x_i; c_o, \ldots, c_n] \frac{\partial F}{\partial c_j} \right\}, \end{aligned}$$

como habíamos visto la condición necesaria de extremo da lugar al sistema de las ecuaciones normales, que en este caso será no lineal

$$\sum_{i=o}^{N} \left\{ [f(x_i) - F(x_i; c_o^*, \ldots, c_n^*] \frac{\partial F}{\partial c_j} \right\} = 0, \quad 0 \leq j \leq n.$$

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:

$$f(x) = 0, \quad x = (x_1, x_2, \ldots, x_m)^T, \quad f : \Re^m \longrightarrow \Re^n,$$

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 45 *Aproximar la función tabulada f:*

x	x_o	x_1	$\cdots$	x_N
$f(x)$	f_o	f_1	$\cdots$	f_N

por una función exponencial mínimo cuadrática de la forma $F(x) = c_o e^{c_1 x}$ *, sin linealizar.*

Aplicando la forma general del método de los mínimos cuadrados, definimos

$$S = \sum_{i=o}^{N} [f(x_i) - c_o e^{c_1 x_i}]^2.$$

Derivando con respecto a los c_j e igualando a cero, se obtien el SEN:

$$\begin{aligned} \sum_{i=o}^{N} [\exp(c_1^* x_i) f_i - c_o^* \exp(2c_1^* x_i)] &= 0 \\ \sum_{i=o}^{N} x_i [\exp(c_1^* x_i) f_i - c_o^* \exp(2c_1^* x_i)] &= 0. \end{aligned}$$

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 $E = \sqrt{\sum_{i=o}^{3} [f_i - \widehat{F}_i]^2}$:

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, pudiendo obtenerse 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 : $g \, \epsilon \, C_{[a,b]}$

$$\left\| g \right\|_2 = \sqrt{\int_a^b g(x)^2 dx} : \quad \text{norma euclideana}$$

$$\left\| g \right\|_\infty = \max_{x \epsilon [a,b]} |g(x)| : \quad \text{norma de Chebyshev}$$

Las dos normas son caso especial 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 correspondiente 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 $\omega = (\omega(x_1), ..., \omega(x_m))$. Se obtienen entonces las expresiones:

$$\|g\|_{p,\omega} = \left(\int_a^b \omega(x) \, |g(x)|^p \, dx \right)^{1/p}$$

$$\|g\|_{p,\omega,M} = \left(\sum_{i=1}^m \omega(x_i) \, |g(x_i)|^p \right)^{1/p}$$

d) Producto escalar

El producto escalar $< g, h >$ de g y h pertenecientes a L 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:

$$< g, g >\geq 0 \quad \forall g, \text{ y } \quad < g, g >= 0 \Longrightarrow g = 0$$

$$< g, h >=< h, g > \text{ : conmutativa}$$

$$< \alpha g, h >= \alpha < g, h > \text{ :asociativa con respecto a multiplicación por un escalar}$$

$$< g + f, h >=< g, h > + < f, h > \text{ : distributiva}$$

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, g y h ϵL se dice que son ortogonales si se cumple que $< g, h >= 0$.

4.2.2. Algunos resultados teóricos generales

Definición 46 *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 47 *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: $\frac{1}{2}\varphi(x) + \frac{1}{2}\varphi(y) - \varphi(\frac{x+y}{2}) \geq \gamma \|x - y\|^2$, para $x, y \in B$, $\gamma > 0$.*

Teorema 48 *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 49 *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.*