Análisis Numérico

Facultad de Ciencias

Kay Tucci

kay@ula.ve

SUMA

Facultad de Ciencias

Universidad de Los Andes (ULA)

Mérida, 5101 - VENEZUELA

Programa del Curso

- Fuente y propagación de errores
- Solución de ecuaciones no lineales de una variable
- Solución de sistemas de ecuaciones lineales
- Teoría de interpolación
- Integración numérica

Bibliografía y Materiales

Bibliografía

- Kincaid D. & Cheney W. *An Introduction to Numerical Analisys*
- Atkinton Kendall E. *An Introduction to Numerical Analisys*
- Burden R. y Faires D. *Análisis Numérico*
- Trevisan M. C. *Notas de Análisis Numérico*

Materiales

- `http://www.matematica.ula.ve/publicaciones/index.html`
- `http://webdelprofesor.ula.ve/ciencias/kay`

Recomendaciones Generales

Algunos conceptos

Problema numérico: Descripción precisa de la relación funcional entre un conjunto finito de datos de entrada y un conjunto finito de datos de salida.

Algoritmo: secuencia ordenada y finita de pasos, excenta de ambigüedades, que seguidas en su orden lógico nos conduce a la solución de un problema específico

Método numérico: Procedimiento para transformar un problema matemático en numérico y resolver este último

Pasos Generales

El análisis numérico se utiliza generalmente cuando no se puede resolver el problema matemático, es decir hallar una relación funcional entre el conjunto de entrada y el de salida. Los pasos a seguir son:

1. Estudio teórico del problema: existencia y unicidad de la solución.
2. Aproximación: Crear una solución para un número finito de valores

   existencia y unicidad

   estabilidad y convergencia
3. Resolución: Elección de un algoritmo numérico

   Elección del algoritmo: Costo y estabilidad

   Codificación del algoritmo

   Ejecución del programa

Fuente y propagación de errores

- Sistemas numéricos
- Aritmética del computador
- Fuentes de errores
- Errores de redondeo y discretización
- Propagación de errores
- Estabilidad e inestabilidad numérica

Sistemas numéricos

Los sistemas numéricos más antiguos son:

- Babilónico: base 60
- Romano: (I, V, X, L, C, D y M)
- Hindú y árabe: decimal

El extendido uso del sistema decimal oculta la existencia de otros sistemas numéricos:

- Binario: base 2
- Octal: base 8
- Hexadecimal: base 16
- . . .

Sistemas numéricos de posición

Los sistemas númericos actuales, decimal, binario, octal, hexadecimal, entre otros; representan a los números reales mediante un sistema de posición con base b.

$$\pm x = \pm(a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots) = \pm \sum_{i=-\infty}^{n} a_i b^i \ ;$$

donde, a es el valor absoluto del número real a representar, $0 \leq a_i \leq b-1$ en el dígito que ocupa la posición i en a contadas a partir del punto decimal, positivos a la izquierda y negativos a la derecha; y n en la posición más a la izquierda ocupada por un dígito $a_n \neq 0$

Ambigüedades:

$$a.ccccc\cdots = a+1 \quad ; \quad c = b-1$$

Sistema numéricos de posición

Conversión de base 10 a base b: convertir el número 12.3125_{10} a base 2

1. Dividir, sucesivamente, la parte entera del número (12_{10}), entre la nueva base (2), hasta obtener un cociente más pequeño que esta última (2), asignando sucesivamente los restos de la división entera a $a_0 = 0, a_1 = 0, a_2 = 1, a_3 = 1$.

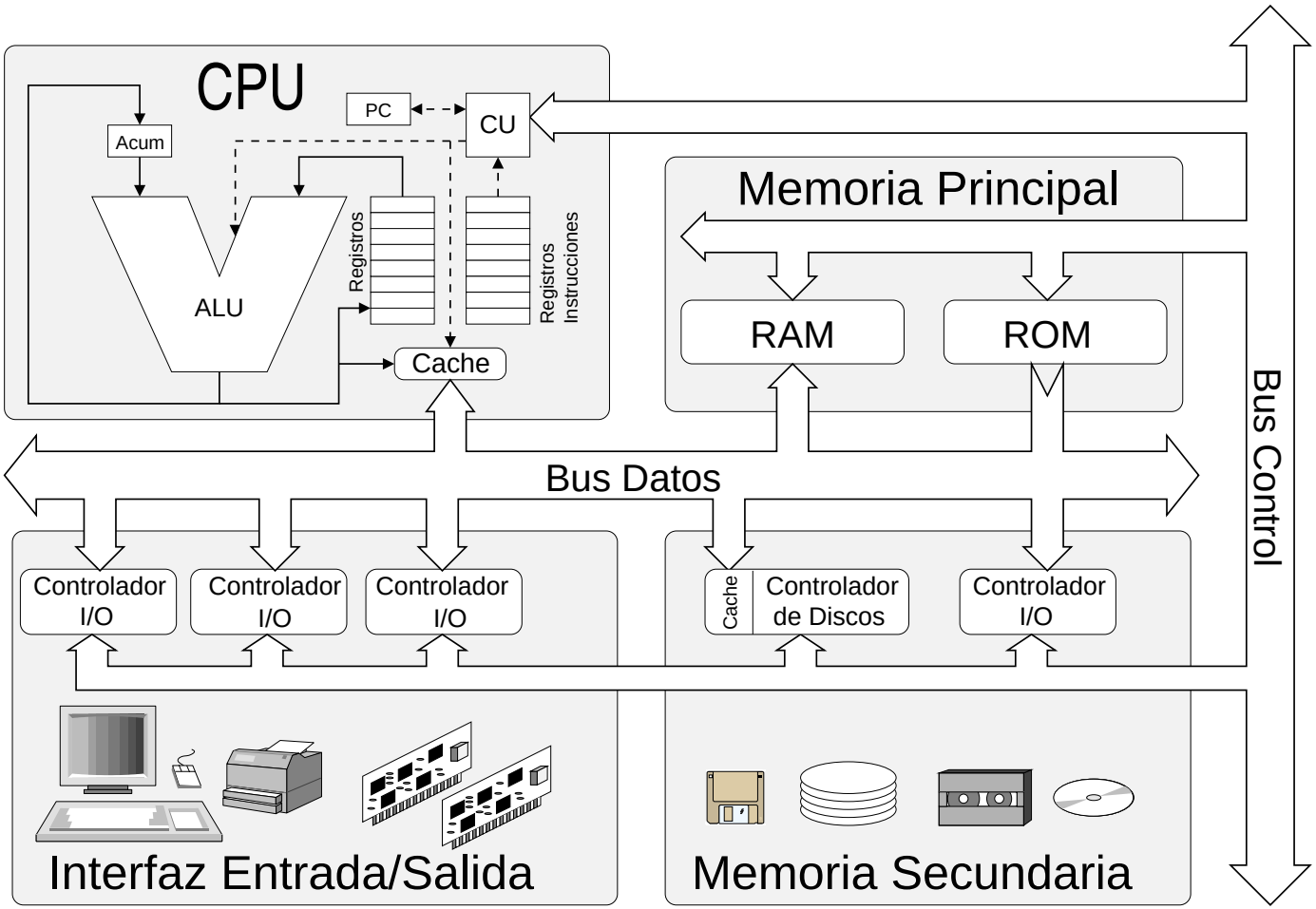
$$12_{10} = 1100_2$$

2. Multiplicamos la parte fraccionaria del número decimal (0.3125_{10}) por la nueva base b (2), restando y asignando sucesivamente la parte entera del producto a $a_{-1} = 0, a_{-2} = 1, a_{-3} = 0, a_{-4} = 1$

$$0.3125_{10} = 0.0101_2$$

$$12.3125_{10} = 1100.0101_2$$

El computador



Sistema de numeración binario

Utiliza solamente dos dígitos o símbolos: “0” y “1”, y su base es 2.

Definiciones:

bit: del ingles *Binary digIT*, representa un dígito binario. Ademas, el bit es la unidad mínima de información empleada en la teoría de la información.

byte u octeto: Generalmente se refiere a 8 bits. Formalmente es una secuencia de bits contiguos, cuyo tamaño depende del código de información o de carácteres en que esté usando.

Sistema de numeración binario

Nombre	Abrev.	Factor	Valor
kilo	K	2^{10}	1024
mega	M	2^{20}	1048576
giga	G	2^{30}	1073741824
tera	T	2^{40}	1099511627776
peta	P	2^{50}	1125899906842624
exa	E	2^{60}	1152921504606846976
zetta	Z	2^{70}	1180591620717411303424
yotta	Y	2^{80}	1208925819614629174706176
bronto	B	2^{90}	1237940039285380274899124224
geop	Ge	2^{100}	1267650600228229401496703205376

Sistema de numeración binario

Existen variantes del sistema de numeración binario

Binario puro: Solamente se pueden representar números no negativos

Signo Magnitud: El bit más significativo representa el signo ($0 = +$) y ($1 = -$) y los restantes $n - 1$ bits representan el valor absoluto del número.

Complemento a la Base Disminuida: En los números negativos cada dígito se escribe como $[d]_b = (b - 1) - (d_i)_b$

Complemento a la Base: $[N]_b$ de un número $(N)_b$ se define como:

$$[N]_b = b^n - (N)_b,$$

donde, n es el número de dígitos de $(N)_b$

Sistema de numeración binario

En 1985 la IEEE establece el *Binary Floating Point Arithmetic Standard* 754-1985, donde se establecen los formatos para representar números punto flotantes de precisión simple (32 bits) y doble (64 bits)

Los números se distribuyen de forma exponencial en la recta real

$$(-1)^S 2^{E-127}(1+0.F) \quad | \quad (-1)^S 2^{E-1023}(1+0.F)$$

donde, S representa el bit de signo, E el exponente y F la parte fracción binaria del número.

presición	signo	exponente		mantisa	
simple	1	8 bits	$10^{-39} \leftrightarrow 10^{38}$	23 bits	7 cifras dec.
doble	1	11 bits	$10^{-308} \leftrightarrow 10^{308}$	52 bits	16 cifras dec.

Desbordamiento

Los resultados que en valor absoluto son menores que el valor mínimo que soporta el formato de representación son considerados *underflows*, mientras los que son mayores que el valor máximo permitido se consideran *overflows*.

```
x ← 1 ; i ← 0
Repita
| ant ← x
| i ← i + 1
| x ← 2*x
| Escribir(i, ant, x)
hasta(ant > x)
```

```
x ← 1
i ← 0
Mientras (1 + x > 1 )
| x ← x / 2
| i ← i + 1
| Escribir(i, x)
```

Fuentes de errores

- En los datos de entrada
- En los códigos
- *Temperamento del computador:*
  “El programa no quiere correr”
- Cuantización del sistema de numeración
- de redondeo o truncamiento:

$$2\left(\tfrac{1}{3}\right) - \tfrac{2}{3} \neq 0.6666666 - 0.6666667 = -0.0000001$$

- Aproximación en los algoritmos:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \cong \sum_{n=0}^{N} \frac{x^n}{n!}$$

Errores relativo y absoluto

Sea x^* el valor aproximado de una cantidad x

- **Error absoluto**: $|x^* - x|$
- **Error relativo**: $\left|\frac{x^*-x}{x}\right|, x \neq 0$
- m es una **cota del error absoluto** si: $m > 0$ y $|x^* - x| \leq m$
- m es una **cota del error relativo** si: $m > 0$ y $\left|\frac{x^*-x}{x}\right| \leq m, x \neq 0$

Cifras significativas

Decimos que a es la representación normalizada de un número real no nulo x si

$$x = \pm 0.a_1 a_2 a_3 \cdots \times 10^e \ ,$$

donde, a_1 es un dígito del 1 al 9; $a_i, i = 2, 3, \ldots$, son dígitos del 0 al 9 y $e \in \mathbb{Z}$ es el exponente.

- a^* aproxima a a con t **decimales correctos** si t es el mayor entero no negativo para el cual: $|a^* - a| \leq 0.5 \times 10^{-t}$
- a^* aproxima a a con t **cifras significativas** si t es el mayor entero no negativo para el cual: $|a^* - a| < 0.5 \times 10^{e-t}$

donde, a^* el valor aproximado de una cantidad a

Propagación de errores

Sean x^*, y^* los valores aproximados de x, y; y ϵ_x, $\epsilon_y \ll 1$ sus respectivos errores relativos.

Suma / Resta :

$$\epsilon_{x\pm y} = \left|\frac{(x^* \pm y^*) - (x \pm y)}{x \pm y}\right| = \left|\frac{x}{x \pm y}\epsilon_x \pm \frac{y}{x \pm y}\epsilon_y\right|$$

Multiplicación :

$$\epsilon_{xy} = \left|\frac{(x^* y^*) - (xy)}{xy}\right| = \epsilon_x + \epsilon_y + \epsilon_x\epsilon_y \cong \epsilon_x + \epsilon_y$$

División :

$$\epsilon_{x/y} = \left|\frac{(x^*/y^*) - (x/y)}{x/y}\right| = \left|\frac{\epsilon_x - \epsilon_y}{1 + \epsilon_y}\right| \cong \epsilon_x - \epsilon_y$$

Propagación de errores

Para evitar la propagación rádida de los errores hay que tener en cuenta que en la aritmética del computador la propiedad asosiativa no se cumple.

- Resta de números muy parecidos

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} = \frac{-2c}{b + \sqrt{b^2 - 4ac}}$$

- Sumatorias largas de números

$$\bar{x} = \frac{1}{10^5} \sum_{i=1}^{10^5} x_i = \frac{1}{100} \sum_{j=0}^{99} \left(\frac{1}{100} \sum_{i=1}^{100} x_{i+100j} \right)$$

Ordenar los terminos de menor a mayor

Error total e iteraciones óptimas

Generalmente en los algoritmos iterativos los errores de aproximación dependen del número de iteraciones N y pueden modelarse como,

$$\epsilon_{aprox} \simeq \frac{\alpha}{N^{\beta}},$$

y los errores de redondeo dependen del precisión de la máquina ϵ_{maq} y van como

$$\epsilon_{red} \simeq \sqrt{N}\epsilon_{maq}$$

Entonces, el error total ϵ_t viene dado por,

$$\epsilon_t = \epsilon_{aprox} + \epsilon_{red} \simeq \frac{\alpha}{N^{\beta}} + \sqrt{N}\epsilon_{maq}$$

El número de iteraciones óptima se alcanza cuando ϵ_t sea mínimo

Cálculo del error

Cómo se propagan el errores de los datos **x** cuando se calcula $f(\mathbf{x})$

$$\epsilon_f \cong \sum_i \left| \frac{\partial f(\mathbf{x})}{x_i} \frac{x_i}{f(\mathbf{x})} \right| |\epsilon_{x_i}|$$

Empíricamente, comparamos los valores calculados con N y $2N$ iteraciones. Cuando se alcanza el valor asimtótico y el error de redondeo no es significativo se tiene que,

$$f_N(\mathbf{x}) - f_{2N}(\mathbf{x}) \simeq \frac{\alpha}{N^\beta},$$

En un gráfico $\log_{10}(f_N - f_{2N})$ vs $\log_{10}(N)$ esto es una recta.

En la gráfica anterior, el punto N_o en el que cambia la pendiente indica aproximadamente el número óptimo de iteraciones, y el $\log_{10}(N_o)$ aproxima al número de cifras significativas del resultado numérico.

Símbolos O y o

- Decimos que f es de orden O *grande* con respecto a g cuando $x \to a$ si existe una constante $K > 0$ tal que,

$$\left| \frac{f(x)}{g(x)} \right| \leq K,$$

para $x \neq a$ en un entorno de a.

Pondremos $f = O(g)$ cuando $x \to a$.

- Decimos que f es de orden o *pequeña* con respecto a g cuando $x \to a$ si,

$$\lim_{x \to a} \frac{f(x)}{g(x)} = 0.$$

Pondremos $f = o(g)$ cuando $x \to a$.

Símbolos O y o

Ejemplo:

- $f = O(1)$ cuando $x \to a$ significa que $f(x)$ se mantiane acotada cuando $x \to a$
- $f = o(1)$ cuando $x \to a$ significa que $f(x)$ tiende a cero cuando $x \to a$

Propiedades:

- Si $f = o(1)$ cuando $x \to a$ entonces $f = O(1)$ cuando $x \to a$
- Si $f = O(x^n)$ cuando $x \to 0$ para algún $n \in \mathbb{Z}^+$ entonces $f = O(x^{n-1})$ cuando $x \to 0$

Estabilidad y Condición

Los algorítmos numéricos debe dar resultados **precisos** y **exactos**.

Estabilidad: Un algoritmo es **estable** si cambios pequeños en los datos iniciales producen cambios pequeños en los resultados

Condición: Algunos algoritmos son estables solamente para un conjunto de condiciones iniciales. Estos algoritmos son **condicionalmente estables**

Crecimiento del error: Si ϵ_0 denota el error inicial y ϵ_n el error despues de n iteraciones, decimos que el error crece:

- **linealmente** si $\epsilon_n \approx kn\epsilon_0$
- **exponencialmente** si $\epsilon_n \approx k^n\epsilon_0$ Hay que evitarlo

Por lo general es inevitable el crecimiento lineal del error.

Ecuaciones no Lineales

- Introducción
- Método de bisección
- Método de la secante
- Método de Newton
- Métodos iterativos de un punto
- Raices múltiples
- Cálculo de raíces de polinomios

Ecuaciones no Lineales

En las Matemáticas aplicada frecuentemente se debe hallar soluciones de la ecuación

$$\mathbf{f}(\mathbf{x}) = 0 \ ,$$

donde, $\mathbf{x} \in \mathbb{R}^n$ y $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ es una función no lineal. En particular, en este curso nos centraremos en la busqueda de las raices o ceros de funciones unidimensionales, $m = 1$, y monovaluadas $n = 1$; es decir,

$$f(x) = 0 \ ,$$

donde, $x \in \mathbb{R}$ y $f : \mathbb{R} \to \mathbb{R}$

Ecuaciones no Lineales

Ecuaciones algebraicas:

- $x^2 - x + 2 = 0$
- $x^6 = x - 1$

Ecuaciones tracendentes:

- $\sin(x) - \frac{x}{2} = 0$
- $\tan(x) = e^x$

Sólo existen soluciones exactas en casos muy particulares y generalmente, se tendrán que utilizar métodos numéricos, en particular, metodos iterativos.

Metodo Iterativo

Es aquel método numérico en el que partiendo de un valor x_0 arbitrario, se calcula una sucesión $x_0, x_1, x_2, \ldots$ de forma recurrente, mediante una relacion de la forma

$$x_{n+1} = g(x_n) \ , n = 0, 1, 2, \ldots \ ;$$

donde, $x_i \in \mathbb{R}$ y $g : \mathbb{R} \to \mathbb{R}$

Los métodos iterativos también se utilizan en otros problemas numéricos y, en general, son muy poco vulnerables al crecimiento del error por redondeo

Iteración: Los pasos que se dan, en un algoritmo, para calcular un iterado, x_{n+1}, a partir del iterado anterior, x_n.

Orden de convergencia

Sea $\{x_n\}_{n=0}^{\infty}$ una sucesión que converge a α y $\epsilon_n = x_n - \alpha$.

- Si existe una constante $C \in (0,1)$ tal que

$$\lim_{n\to\infty} \frac{|\epsilon_{n+1}|}{|\epsilon_n|} = C \,,$$

  entonce, la sucesión **converge** a α linealmente con una **tasa o velocidad de convergencia** C

Si lo anterior ocurre para $C = 0$, entonces se dice que la sucesión converge superlinealmente.

Orden de convergencia

Sea $\{x_n\}_{n=0}^{\infty}$ una sucesión que converge a α y $\epsilon_n = x_n - \alpha$.

- Si existe un $p \geq 1$ y una constante C tal que

$$\lim_{n\to\infty} \frac{|\epsilon_{n+1}|}{|\epsilon_n|^p} = C \ ,$$

entonce, $\{x_n\}_{n=0}^{\infty}$ **converge** a α con **orden** p.

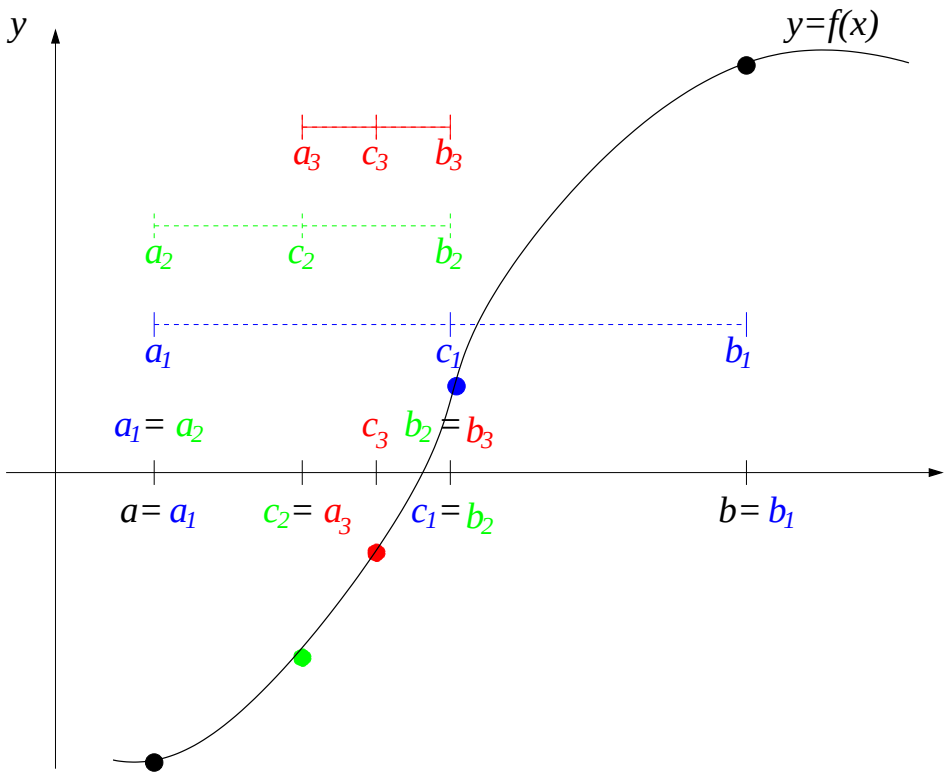
En particular, la convergencia con orden $p = 2$ se le llama cuadrática y la con orden $p = 3$ cúbica

En general los métodos que generen sucesiones con alto orden de convergencia se aproximan más rápidamente a la solución que aquellos que generen sucesiones con bajo orden.

Método de bisección

Entrada: a, b, t, n
Salida: c, 'error'

```
leer(a, b, t, n)
i ← 1
fa ← f(a)
repita
| c ← a+(b-a)/2
| fc ← f(c)
| i ← i + 1
|si (sign(fa)∗sign(fc)>0)
| | fa ← fc ; a ← c
| sino
| | b ← c
hasta (i≥n)∨(fc=0)∨(|b-a|/2≤t)
si (fc=0)∨(|b-a|/2≤t)
| escribir(c)
sino
| escribir('error')
```



Método de bisección

Convergencia

Como $f(a)f(b) \leq 0$ sabemos que existe un $\alpha \in [a, b]$ tal que $f(\alpha) = 0$. Partiendo del algoritmo y mediante inducción se puede ver que

$$b_{n+1} - a_{n+1} = \frac{1}{2}(b_n - a_n) \quad \text{y que} \quad b_n - a_n = \frac{1}{2^{n-1}}(b_1 - a_1)$$

Como $\alpha \in [a_n, c_n]$ o $\alpha \in [c_n, b_n]$ tenemos que

$$|c_n - \alpha| \leq a_n - c_n = c_n - b_n = \frac{1}{2}(a_n - b_n)$$

Combinando las dos equaciones anteriores tenemos que

$$|c_n - \alpha| \leq \frac{1}{2^{n-1}}(b_1 - a_1) \quad , \quad n \geq 1$$

Lo que implica que

$$\lim_{n \leftarrow \infty} c_n = \alpha$$

es decir, convergencia lineal, $p = 1$ con una velocidad de convergencia $C = \frac{1}{2}$

Método de bisección

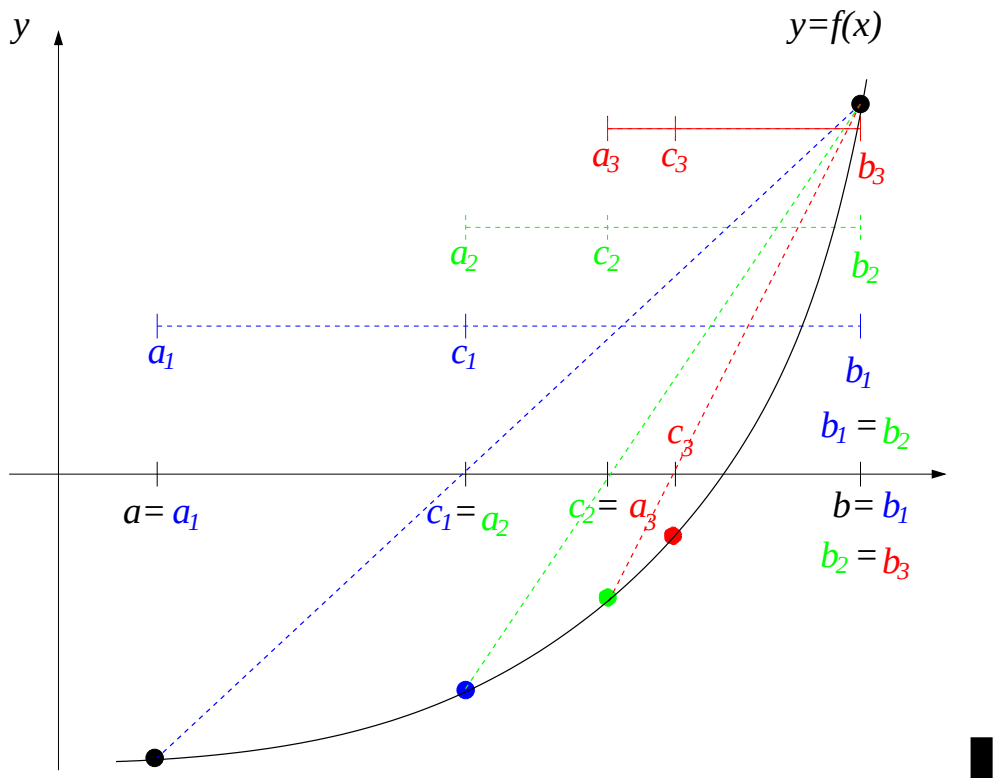
Comentarios del método de bisección:

- Hay que dar dos valores iniciales, uno a cada lado de la raiz que se está buscando. Esto es un problema, especialmente si no se tiene ninguna idea del comportamiento de la función o si esta presenta raices muy similares o múltiples.
- Errores de redondeo o de cuantificación pueden causar problemas cuando el intervalo de búsqueda se vuelve pequeño
- Al evaluar la función cerca de la raiz, $f(x) \approx 0$, para el cálculo del cambio de signo, los errores por operar con números muy pequeños pueden hacerse presentes

Método de la posición falsa

Entrada: a, b, t, n
Salida: c, 'error'

```
leer(a, b, t, n)
i ← 0; c ← a
fa ← f(a); fb ← f(b)
repita
| i ← i + 1; ant ← c
| c ← b-fb∗(b-a)/(fb-fa)
| fc ← f(c)
|si (sign(fa)∗sign(fc)>0)
| | fa ← fc ; a ← c
| sino
| | fb ← fc ; b ← c
hasta (i≥n)∨(|c-ant|≤t)
si (|c-ant|≤t)
| escribir(c)
sino
| escribir('error')
```



Método de la posición falsa

Convergencia

Diferencias divididas:

Definición: Sean $a \neq b \neq c$,

diferencia dividida de primer orden: $f[a,b] = \dfrac{f(b)-f(a)}{b-a}$

diferencia dividida de segundo orden: $f[a,b,c] = \dfrac{f[b,c]-f[a,b]}{c-a}$

Propiedades:

- $f[a,b] = f[b,a]$
- $f[a_1,a_2,a_3] = f[a_i,a_j,a_k]$ con $i \neq j \neq k$ y $i,j,k \in \{1,2,3\}$
- si f es continua en $[a,b]$ y derivable en (a,b), existe $\xi \in (a,b)$ tal que $f[a,b] = f'(\xi)$ (Teorema del valor medio)
- si f' es derivable en $(min(a,b,c), max(a,b,c))$, existe η tal que $f[a,b,c] = f''(\eta)$

Método de la posición falsa

Convergencia Del método tenemos que:

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

donde el error se calcula como

$$c - \alpha = b - \alpha - \frac{b-a}{f(b)-f(a)} = \frac{1}{2}(a-\alpha)(b-\alpha)\frac{f''(\eta)}{f'(\xi)} \ , \quad \eta, \xi \in (a,b)$$

Teorema: Sea f dos veces continuamente diferenciable en $[a,b]$ con α la única raiz en $[a,b]$. Suponiendo que $f(a)f(b) < 0, f'(\alpha) \neq 0$ y f'' no cambia de signo en $[a,b]$. Si

$$C = \left|\frac{\omega - \alpha}{2}\right| max_{x\in[a,b]} \left|\frac{f''(x)}{f'(x)}\right| < 1 \ , \text{con } \omega = a \text{ o } \omega = b \ ,$$

según sea caso del estremo del intervalo $[a,b]$ que no se modifique, entonces el método converge linealmente.

$$a_{n+1} - \alpha \leq C(a_n - \alpha) \ , \text{si } \omega = b \quad \text{o} \quad b_{n+1} - \alpha \leq C(b_n - \alpha) \ , \text{si } \omega = a$$

Método de la posición falsa

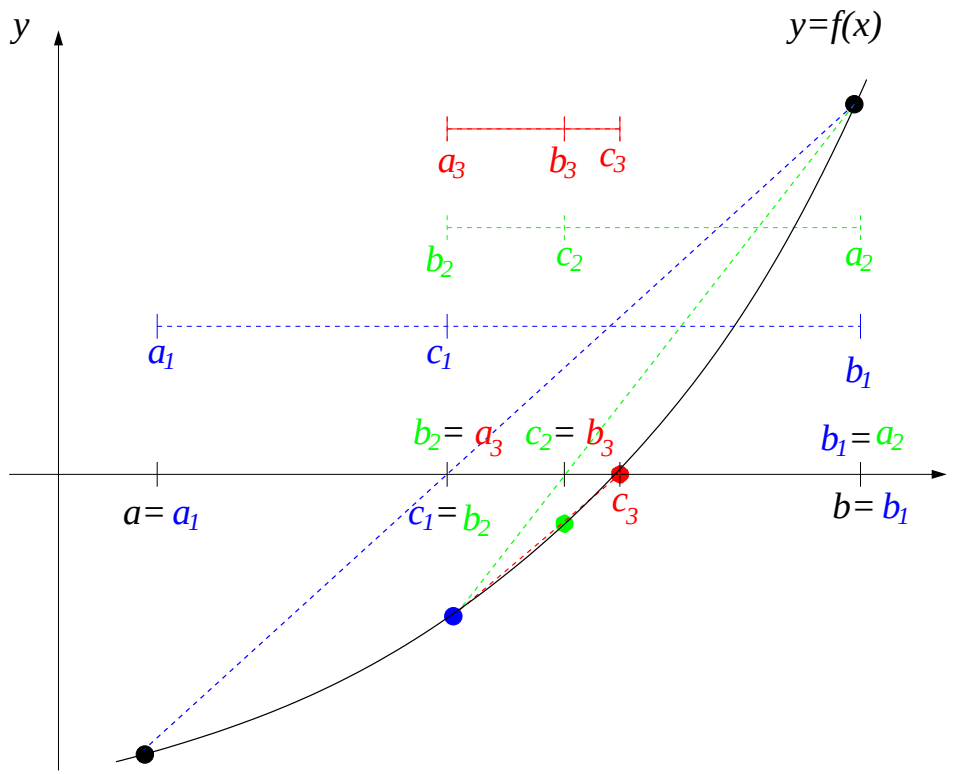
Comentarios del método de la posición falsa:

- Hay que dar dos valores iniciales, uno a cada lado de la raiz que se está buscando. Esto es un problema, especialmente si no se tiene ninguna idea del comportamiento de la función o si esta presenta raices muy similares o múltiples.
- Si la función es convexa o cóncava en $[a, b]$ uno de los extremos de intervalo no se moverá por lo que el método solamente aproxima a la raiz por uno de los lados.
- Al evaluar la función cerca de la raiz, $f(x) \approx 0$, para el cálculo del cambio de signo, los errores por operar con números muy pequeños pueden hacerse presentes

Método de la secante

Entrada: a, b, t, n
Salida: c, 'error'

```
leer(a, b, t, n)
i ← 1
fa ← f(a)
fb ← f(b)
repita
| c ← b-fb*(b-a)/(fb-fa)
| i ← i + 1
| a ← b ;   fa ← fb
| b ← c ;   fb ← f(c)
hasta (i≥n)∨(fc=0)∨(|b-a|/2≤t)
si (fc=0)∨(|b-a|/2≤t)
| escribir(c)
sino
| escribir('error')
```



Método de la secante

Convergencia Del método tenemos que:

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

donde, usando diferencias divididas, el error se calcula como

$$c - \alpha = b - \alpha - \frac{b-a}{f(b)-f(a)} = \frac{1}{2}(a-\alpha)(b-\alpha)\frac{f''(\eta)}{f'(\xi)} \ , \quad \eta, \xi \in (a,b)$$

sustituyendo $c = x_{n+1}, b = x_n, a = x_{n-1}$ tenemos que

$$\epsilon_{n+1} = \epsilon_n \epsilon_{n-1} \frac{f''(\eta_n)}{2f'(\xi_n)}$$

Teorema: Supongamos que $f : \mathbb{R} \to \mathbb{R}$ es una función C^2 en una vecindad de α para la cual $f(\alpha) = 0$ y $f'(\alpha) \neq 0$. Entonces si x_0 y x_1 se seleccionan suficientemente cerca de α las iteraciones del método de la secante convergen a α. Además,

$$\lim_{n\to\infty} \frac{\epsilon_{n+1}}{(\epsilon_n)^p} = \left(\frac{f''(\alpha)}{2f'(\alpha)}\right)^{p-1} \quad , \quad p = \frac{1+\sqrt{5}}{2} \approx 1.62$$

Método de la secante

Demostración de la convergencia: Sea

$$M = \max\left(\frac{f''(x)}{2f'(x)}\right) \quad \forall \quad x \in I = [\alpha - \epsilon, \alpha + \epsilon]\ , \epsilon > 0$$

y supongamos que x_0 y x_1 son escogidos de forma tal que

$$\delta = max\{M|\epsilon_0|, M|\epsilon_1|\} < 1$$

Como $|\epsilon_2| \leq M|\epsilon_1||\epsilon_0|$ tenemos que

$$M|\epsilon_2| \leq M^2|\epsilon_1||\epsilon_0| \leq \delta^2 < \delta < 1 \Longrightarrow |\epsilon_2| < \frac{\delta}{M} < \epsilon$$

Suponiendo que $x_i \in I\ , i = 1, 2, \ldots, n$, lo que significa que $M|\epsilon_i| < \delta$, entonces

$$|\epsilon_{n+1}| \leq M|\epsilon_n||\epsilon_{n-1}|$$

$$M|\epsilon_{n+1}| \leq M^2|\epsilon_n||\epsilon_{n-1}| \leq \delta^2 < \delta < 1 \Longrightarrow |\epsilon_{n+1}| < \frac{\delta}{M} < \epsilon$$

lo que indica que $x_n \in I \quad \forall \quad n$

Método de la secante

Demostración de la convergencia:

$$
\begin{array}{rcllcll}
M|\epsilon_0| & \leq & \delta & & & & \\
M|\epsilon_1| & \leq & \delta & & & & \\
M|\epsilon_2| & \leq & M|\epsilon_0|M|\epsilon_1| & \leq & \delta \times \delta & = & \delta^2 \\
M|\epsilon_3| & \leq & M|\epsilon_1|M|\epsilon_2| & \leq & \delta \times \delta^2 & = & \delta^3 \\
M|\epsilon_4| & \leq & M|\epsilon_2|M|\epsilon_3| & \leq & \delta^2 \times \delta^3 & = & \delta^5 \\
M|\epsilon_5| & \leq & M|\epsilon_3|M|\epsilon_4| & \leq & \delta^3 \times \delta^5 & = & \delta^8 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
M|\epsilon_{n+1}| & \leq & M|\epsilon_{n-1}|M|\epsilon_n| & \leq & \delta^{q_{n-1}} \times \delta^{q_n} & = & \delta^{q_{n-1}+q_n}
\end{array}
$$

Se tiene que $\{q_n\}$ es la sucesión de Fibonacci cuya solución positiva es

$$q_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{n+1}$$

Método de la secante

Demostración de la convergencia: Así que

$$|\epsilon_n| \leq \frac{\delta^{q_n}}{M} \text{ con } \delta < 1 \text{ , obtenemos que } \lim_{n\to\infty} |\epsilon_n| = 0$$

Como

$$q_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{n+1} \text{ , } p = \frac{1+\sqrt{5}}{2}$$

entonces

$$(\delta^{q_n})^p = \delta^{q_{n+1}}$$

y calculando la convergencia tenemos que

$$\lim_{n\to\infty} \frac{\epsilon_{n+1}}{(\epsilon_n)^p} = \lim_{n\to\infty} \frac{\frac{1}{M}\delta^{q_{n+1}}}{\left(\frac{1}{M}\delta^{q_n}\right)^p} = \lim_{n\to\infty} \frac{\frac{1}{M}\delta^{q_{n+1}}}{\frac{1}{M^p}\delta^{q_{n+1}}} = M^{p-1}$$

Método de la secante

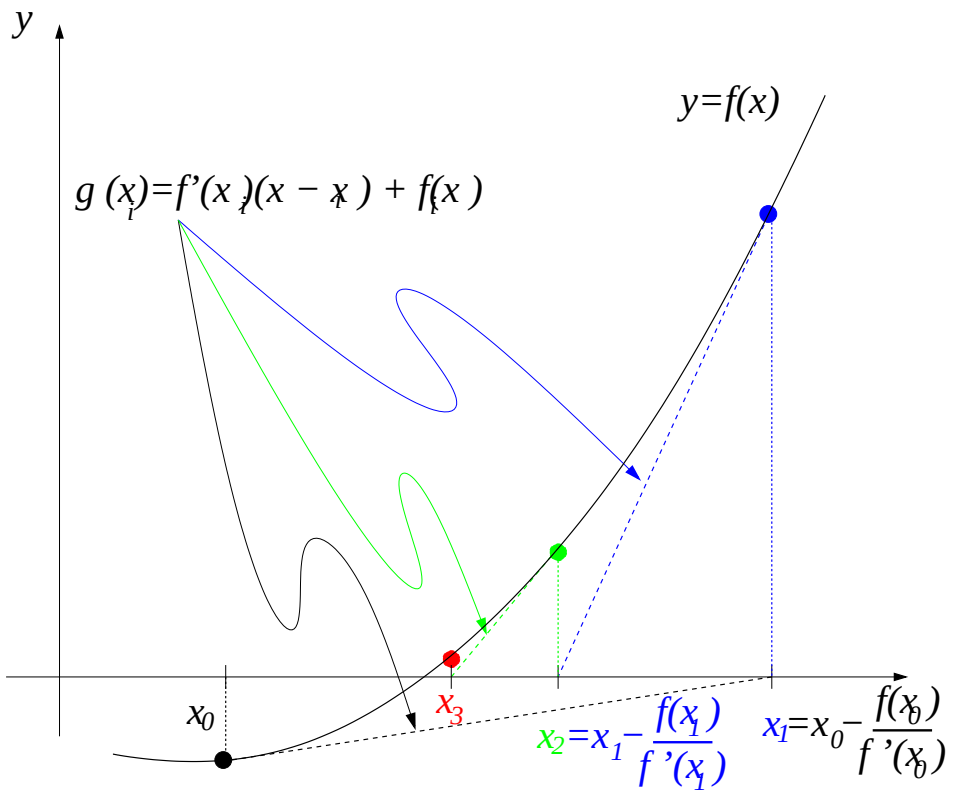
Comentarios del método de la secante:

- Hay que dar dos valores iniciales, a pesar de que no tienen que encerrar a la raiz que se está buscando, los puntos tienen que estar suficientemente cerca de la raiz para garantizar que el método converja.
- Si la derivada de la función, cerca de la raiz, tiene un valor muy alto el cálculo de $b - a$ puede causar problemas por perdida de cifras significativas.
- De igual modo, si la derivada de la función, cerca de la raiz, tiene un valor muy bajo el cálculo de $f(b) - f(a)$ puede causar problemas por perdida de cifras significativas

Método de Newton-Raphson

Entrada: x, t, n
Salida: x, 'error'

```
leer(x, t, n)
i ← 0
repita
| i ← i + 1
| xa ← x
| x ← xa - f(xa)/fp(xa)
hasta (i≥n)∨(|x-xa|≤t)
si (|x-xa|≤t)
| escribir(x)
sino
| escribir('error')
```



Método de Newton-Raphson

Convergencia

Del método de Newton-Raphson tenemos que

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad , \quad n \geq 0$$

Expandiendo a $f(x)$ alrededor de x_n:

$$f(x) = f(x_n) + (x - x_n)f'(x_n) + \frac{(x - x_n)^2}{2} f''(\xi_n) \quad , \quad \xi_n \in [x_n, x] \text{ o } [x, x_n]$$

Para $x = \alpha$, $f(\alpha) = 0$ y dividiendo lo anterior entre $f'(x_n)$ tenemos

$$\underbrace{\frac{f(x_n)}{f'(x_n)} + (\alpha - x_n)}_{-\epsilon_{n+1}} + \underbrace{(\alpha - x_n)}_{\epsilon_n}^2 \frac{f''(\xi_n)}{2f'(x_n)} = 0$$

Que es lo mismo que

$$\epsilon_{n+1} = \epsilon_n^2 \frac{f''(\xi_n)}{2f'(x_n)}$$

Método de Newton-Raphson

Convergencia

Teorema: Sea $f \in C^2[a,b]$. Si $\alpha \in [a,b]$ es tal que $f(\alpha) = 0$ y $f'(\alpha) \neq 0$, entonces existe un $\epsilon > 0$ tal que el método de Newton-Raphson genere una sucesión $\{x_n\}_{n=1}^{\infty}$ que converge cuadráticamente a α para cualquier valor inicial $x_0 \in I = [\alpha - \epsilon, \alpha + \epsilon]$

Demostración: Definamos a

$$M = \frac{1}{2} \max_{x \in I} \left| \frac{f''(\xi_n)}{f'(x_n)} \right|$$, y escojamos a $x_0 \in I$ tal que $M|\epsilon_0| < 1$

Sustituyendo en la ecuación de la sucesión $\{\epsilon_n\}$ tenemos

$$|\epsilon_1| \leq (M|\epsilon_0|)^2 < |\epsilon_0| \leq \epsilon$$, lo que implica que $x_1 \in I$

Ahora si suponemos que $x_n \in I$ y que $M|\epsilon_n| < 1$ tenemos

$$|\epsilon_{n+1}| \leq (M|\epsilon_n|)^2 < |\epsilon_n| \leq \epsilon$$, lo que implica que $x_{n+1} \in I$

Método de Newton-Raphson

Demostración de la convergencia: De lo anterior concluimos que

$$x_n \in I \quad , \quad M|\epsilon_n| < 1 \; \forall \; n \quad \text{y} \quad M|\epsilon_{n+1}| \leq (M|\epsilon_n|)^2 \leq (M|\epsilon_0|)^{2n}$$

obteniendo que $|\epsilon_n| = 0$ cuando $n \to \infty$ ya que $M|\epsilon_0| < 1$, lo que equivale a

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

Además,

$$\lim_{n \to \infty} \frac{\epsilon_{n+1}}{\epsilon_n^2} = \frac{1}{2} \frac{f''(\alpha)}{f'(\alpha)}$$

Es decir, el método tiene un orden de convergencia

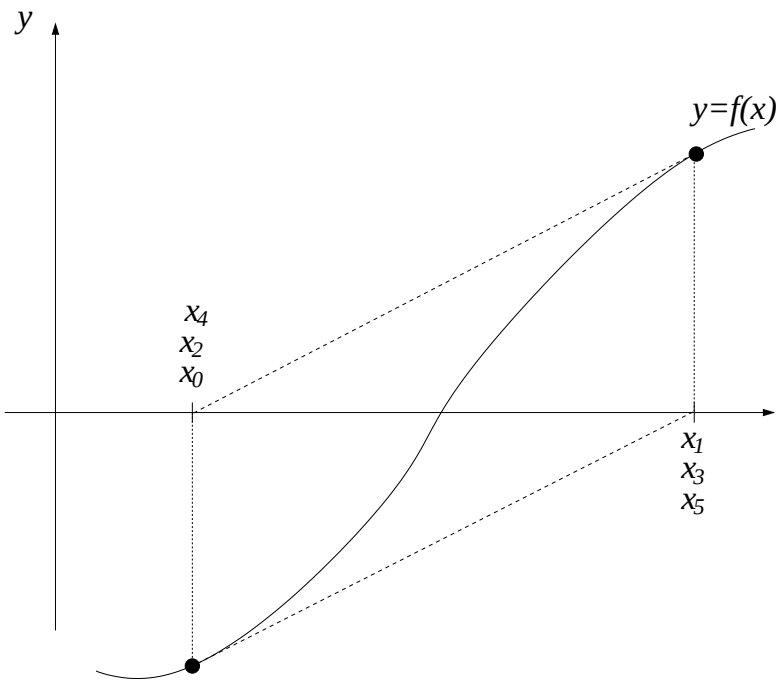
$$p = 2 \quad \text{con} \quad C = \frac{1}{2} \frac{f''(\alpha)}{f'(\alpha)}$$

Método de Newton-Raphson

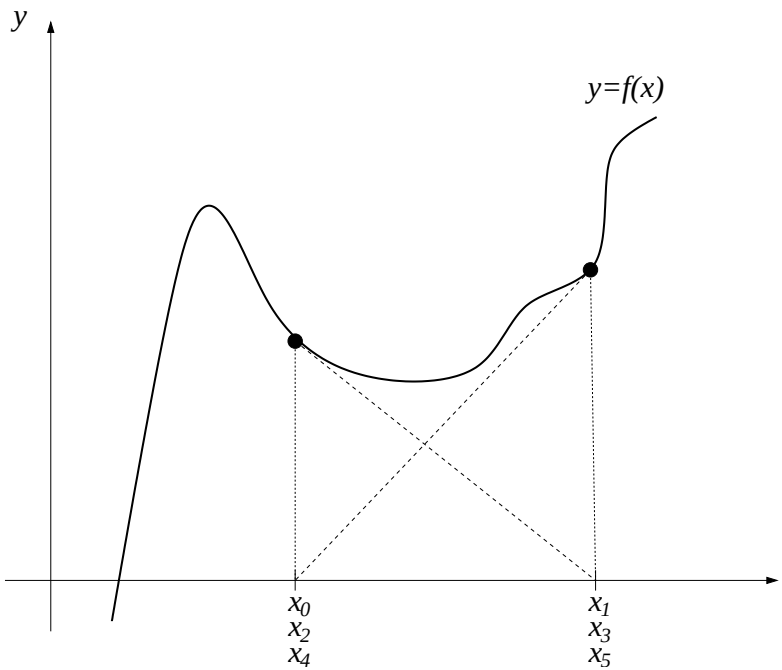
Trampas

Cambio de concavidad

$$f''(x) = 0$$



Raiz no real



Método iterativo de un punto

Partiendo de la ecuación general

$$f(x) = 0 \ ,$$

cuyas raices se quieren encontrar. Supongamos que α es una de sus raices y que existe otro valor x_0 que no satisface la ecuación.

$$f(\alpha) = 0 \ , \quad f(x_0) \neq 0$$

Realizando algunas operaciones algebraicas transformaos la ecuación

$$f(x) = 0 \quad \text{en} \quad g(x) = x \ ,$$

donde, $g(x)$ es una nueva función que cumple con

$$g(\alpha) = \alpha \ , \quad g(x_0) \neq x_0$$

Método iterativo de un punto

Al evaluar un valor cualquiera x_0 en $g(x)$ obtenemos un nuevo valor

$$x_1 = g(x_0) \ ,$$

donde, si el nuevo valor

$$\begin{cases} x_1 = x_0 & \text{, encontramos la raiz } x_0 = \alpha \\ x_1 \neq x_0 & \text{, es lo que generalmente pasa} \end{cases}$$

Si repetimos el proceso obtenemos

$$x_1 = g(x_0) \quad , \ x_2 = g(x_1) \quad , \ \ldots , \ x_n = g(x_{n-1})$$

Método iterativo de un punto

La sucesión $\{x_n\}$ converge a la raiz α si

$$\lim_{n\to\infty}(x_n-\alpha)=0 \quad \text{, es decir,} \quad \lim_{n\to\infty} x_n=\alpha$$

Si la expresión anteriror es cierta, debe valer también para x_{n+1}

$$\lim_{n\to\infty}(x_{n+1}-\alpha)=0 \quad \text{, y como } x_n \text{ tiende a } \alpha \text{ ,} \quad \lim_{n\to\infty}(x_{n+1}-x_n)=0$$

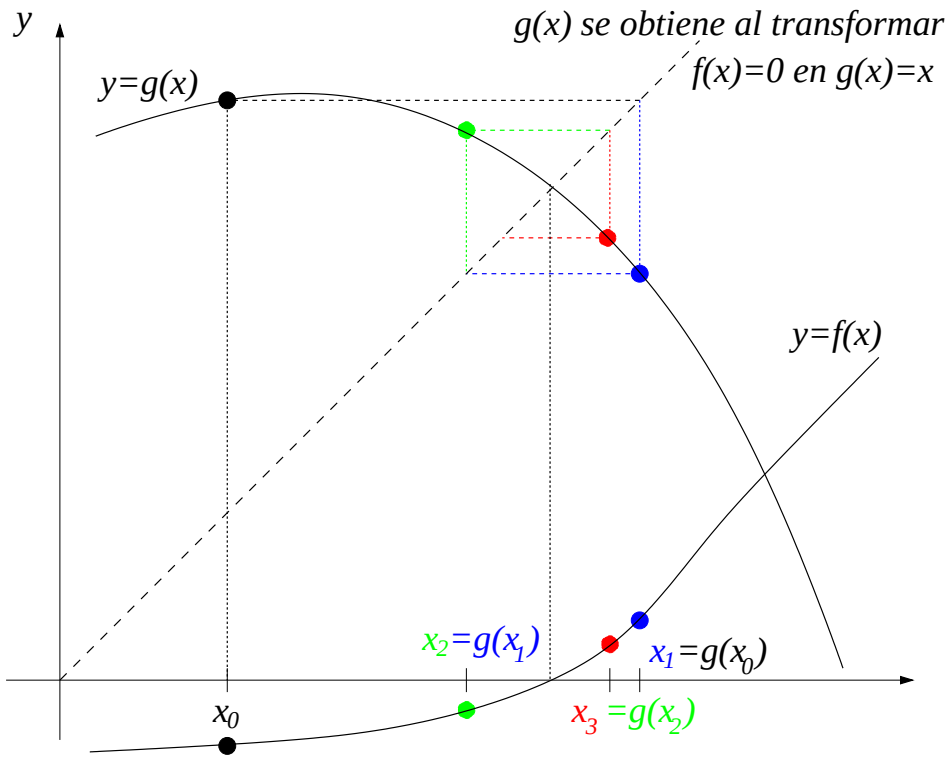
Es decir, cuando el número de iteraciones se vuelve muy grande la diferencia entre dos aproximaciones sucesivas x_n y x_{n+1} de la raiz α debe aproximarse a cero.

Esta condición es necesaria para que los procesos iterativos converjan a la raiz.

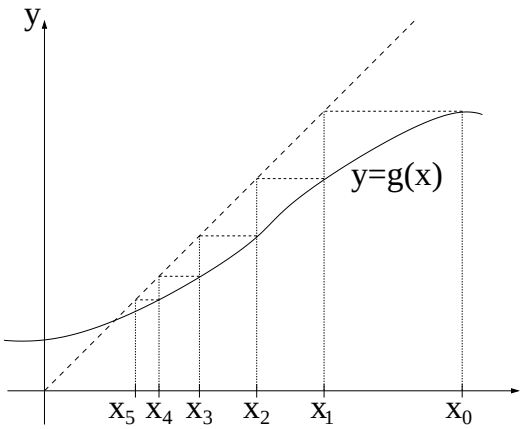
Método iterativo de un punto

Entrada: x, t, n
Salida: x, 'error'

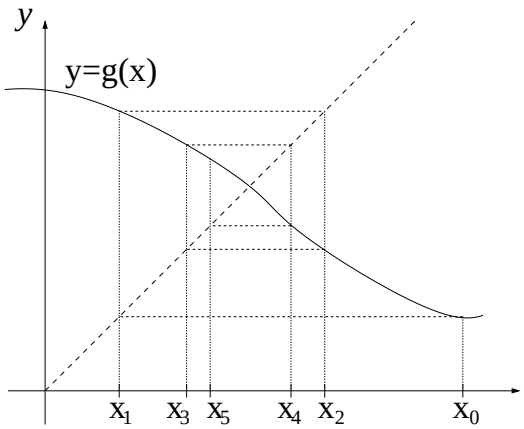
```
leer(x, t, n)
i ← 0
repita
| i ← i + 1
| xa ← x
| x ← g(xa)
hasta (i≥n)∨(|x-xa|≤t)
si (|x-xa|≤t)
| escribir(x)
sino
| escribir('error')
```



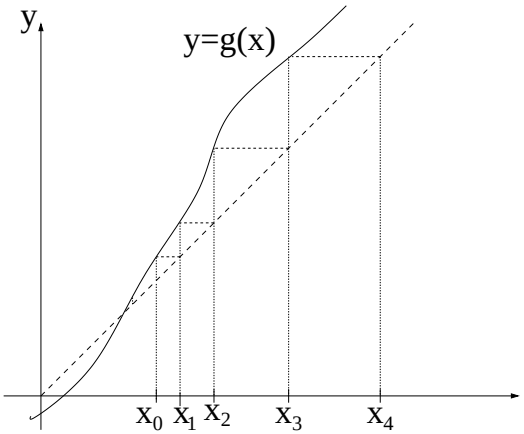
Método iterativo de un punto



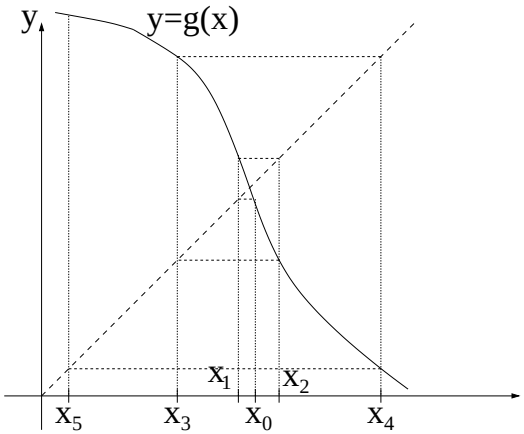
Convergencia Monotónica



Convergencia Oscilante



Divergencia Monotónica



Divergencia Oscilante

Método iterativo de un punto

Criterio de convergencia

Del teorema del valor medio sabemos que

$$g(x_n) - g(x_{n-1}) = g'(\xi_n)(x_n - x_{n-1}) \ ,$$

donde, $\xi_n \in$ región entre x_n y x_{n-1} y $g'(x)$ es la derivada de $g(x)$. Como $g(x_n) = x_{n+1}$ y $g(x_{n-1}) = x_n$ podemos decir

$$|x_{n+1} - x_n| = |g'(\xi_n)||(x_n - x_{n-1})|$$

Si suponemos que $|g'(\xi_n)| \leq M$, es decir está acotada en el entorno de la raiz tenemos que

$$|x_{n+1} - x_n| \leq M|x_n - x_{n-1}| \leq M^2|x_{n-1} - x_{n-2}|$$

Método iterativo de un punto

Criterio de convergencia

Continuando con la sustitucion hasta el termino $M|x_1 - x_o|$ tenemos

$$|x_{n+1} - x_n| \leq M^n |x_1 - x_o|$$

Como sabemos que para que la serie converja $x_{n+1} - x_n$ debe tender a cero, entonces

$$\lim_{n \to \infty} M^n |x_1 - x_o| = 0 \ ,$$

y como $|x_1 - x_o| \neq 0$ a menos que $x_0 = \alpha$ tenemos que

$$|g'(x)| \leq M < 1 \quad \forall \ x \text{ en un entorno de } \alpha \text{ que contenga a } x_0$$

Es condición suficiente, más no necesaria, para la convergencia

Método iterativo de un punto

Criterio de convergencia

Teorema: Sea α una raiz de $x = g(x)$, con la función $g(x)$ $p \geq 2$ veces continuamente diferenciable en un entorno de α. Además, supongamos que

$$g^{(j)}(\alpha) = 0 \quad \forall \quad j < p$$

Si se elige a x_0 suficientemente cerca de α el método iterativo de un punto converge a α con un orden de al menos p y se cumple

$$\lim_{n \to \infty} \frac{\epsilon_{n+1}}{\epsilon_n^p} = \frac{g^{(p)}(\alpha)}{p!}$$

Método iterativo de un punto

Criterio de convergencia

Demostración: Expandiendo a $g(x_n)$ con una serie de Taylor alrededor de α tenemos

$$x_{n+1} = g(\alpha)+(x_n-\alpha)g'(\alpha)+\cdots+\frac{(x_n-\alpha)^{p-1}}{(p-1)!}g^{(p-1)}(\alpha)+\frac{(x_n-\alpha)^p}{(p)!}g^{(p)}(\xi_n)\ ,$$

con ξ_n entre x_n y α. Como $g^{(j)}(\alpha) = 0 \forall j < p$ y $g(\alpha) = \alpha$, obtenemos

$$x_{n+1} - \alpha = \frac{(x_n-\alpha)^p}{(p)!}g^{(p)}(\xi_n)$$

Como el método converge por ser $g'(\alpha) = 0 < 1$, $x_n \to \alpha$ y por lo tanto

$$\lim_{n\to\infty}\frac{x_{n+1}-\alpha}{(x_n-\alpha)^p} = \lim_{n\to\infty}\frac{\epsilon_{n+1}}{\epsilon_n^p} = \frac{g^{(p)}(\alpha)}{p!}$$

Método iterativo de un punto

Raices múltiples: Cuando un método iterativo se aproxima a una raiz de $f(x)$ el orden de convergencia del método se pierde cuando la raiz es múltiple.

Definición: Decimos que α es un cero de multiplicidad m de $f(x)$ si

$$f(x) = (x-\alpha)^m q(x)\ \forall x \neq \alpha \quad \text{, donde,} \quad \lim_{x\to\alpha} q(x) \neq 0$$

Esencialmente $q(x)$ representa la parte de $f(x)$ que no contribuye al cero de la función.

Teorema: La función $f \in C^m[a,b]$ tiene un cero de multiplicidad m en $\alpha \in (a,b)$ ssi

$$f^{(j)}(\alpha) = 0 \quad \forall \quad j < m \quad \text{y} \quad f^m(\alpha) \neq 0$$

Método iterativo de un punto

Raices múltiples: Para resolver este problema se define una función

$$\phi(x) = \frac{f(x)}{f'(x)} \frac{(x-\alpha)^m}{m(x-\alpha)^{m-1}q(x) + (x-\alpha)^m q'(x)}$$

$$\phi(x) = (x-\alpha)\frac{q(x)}{mq(x) + (x-\alpha)q'(x)}$$

que tiene un único cero en α, ya que $q(\alpha) \neq 0$

Luego, aplicamos algún método iterativo usando $\phi(x)$ como sustituta de $f(x)$. Por ejemplo para el método de Newton-Raphson tendríamos que

$$g(x) = x - \frac{\phi(x)}{\phi'(x)} = x - \frac{f(x)f'(x)}{(f'(x))^2 - f(x)f''(x)}$$

Note que se requiere evaluar también a $f''(x)$

Δ^2 de Aitken

El método Δ^2 de Aitken sirve para acelerar la convergencia lineal de una sucesión $\{x_n\}_{n=0}^{\infty}$ a su límite α. Para valores suficientemente grandes de n tenemos que

$$\frac{x_{n+1}-\alpha}{x_n-\alpha} \approx \frac{x_{n+2}-\alpha}{x_{n+1}-\alpha} \quad \rightarrow \quad \alpha \approx x_n - \frac{(x_{n+1}-x_n)^2}{x_{n+2}-2x_{n+1}+x_n}$$

Teorema: Sea una sucesión que converge a α. La sucesión $\{y_n\}$, definida por

$$y_{n+1} = x_n - \frac{(x_{n+1}-x_n)^2}{x_{n+2}-2x_{n+1}+x_n}$$

converge a α más rapidamente que la sucesión $\{x_n\}$, esto es

$$\lim_{n\rightarrow\infty} \frac{y_n-\alpha}{x_n-\alpha} = 0$$

Δ^2 de Aitken

Demostración:

Sustituyendo en la definición los términos $x_n = \epsilon_n + \alpha$ tenemos:

$$y_n = \alpha + \frac{\epsilon_n \epsilon_{n+2} - \epsilon_{n+1}^2}{\epsilon_{n+2} - 2\epsilon_{n+1} + \epsilon_n}$$

Por su parte,

$$\underbrace{x_{n+1} - \alpha}_{\epsilon_{n+1}} = \eta \underbrace{(x_n - \alpha)}_{\epsilon_n} + \delta_n \underbrace{(x_n - \alpha)}_{\epsilon_n}$$

con

$$\lim_{n\to\infty} \frac{x_{n+1} - \alpha}{x_n - \alpha} = \eta \quad , \quad |\eta| < 1 \quad \text{y} \quad \lim_{n\to\infty} \delta_n = 0 \quad ;$$

Δ^2 de Aitken

Demostración:

entonces

$$\epsilon_{n+1} = (\eta + \delta_n)\epsilon_n \quad y \quad \epsilon_{n+2} = (\eta + \delta_{n+1})(\eta + \delta_n)\epsilon_n$$

Sustituyendo esto último en la definición

$$y_n - \alpha = \epsilon_n \frac{(\eta + \delta_{n+1})(\eta + \delta_n) - (\eta + \delta_n)^2}{(\eta + \delta_{n+1})(\eta + \delta_n) - 2(\eta + \delta_n) + 1}$$

por lo que

$$\lim_{n\to\infty} \frac{y_n - \alpha}{x_n - \alpha} = \frac{(\eta + \delta_{n+1})(\eta + \delta_n) - (\eta + \delta_n)^2}{(\eta + \delta_{n+1})(\eta + \delta_n) - 2(\eta + \delta_n) + 1} = 0$$

Método de Steffensen

Resolviendo el problema $f(x) + x = x$ y dado x_0, se calculan

$$x_1 = f(x_0) + x_0 \quad , \quad x_2 = f(x_1) + x_1 \quad , \quad y_1 = x_0 - \frac{(x_1 - x_0)^2}{x_2 - 2x_1 + x_0}$$

para calcular x_3 no utilizamos el valor de x_2 como lo hariamos con el método de punto fijo, sino usamos Δ^2-Aitken ya que suponemos que y_1 es una mejor aproximación de la raiz que x_2,

$$x_3 = y_1 \quad , \quad x_4 = f(x_3) + x_3 \quad , \quad x_5 = f(x_4) + x_4$$

$$y_2 = x_3 - \frac{(x_4 - x_3)^2}{x_5 - 2x_4 + x_3}$$

y así sucesivamente

Método de Steffensen

En la n-esima iteración tendremos que

$$x_{3n} = y_n \quad , \quad x_{3n+1} = f(x_{3n}) + x_{3n} \quad , \quad x_{3n+2} = f(x_{3n+1}) + x_{3n+1} \quad ,$$

$$y_{n+1} = x_{3n} - \frac{(x_{3n+1} - x_{3n})^2}{x_{3n+2} - 2x_{3n+1} + x_{3n}}$$

Sustituyendo para usar la sucesión $\{y_n\}_{n=1}^{\infty}$

$$y_{n+1} = y_n - \frac{(f(y_n) + y_n - y_n)^2}{f(f(y_n) + y_n) + f(y_n) + y_n - (f(y_n) + y_n) + y_n}$$

Simplificando y aplicando en método de punto fijo a y_n tenemos

$$g(x_n) = y_n - \frac{(f(y_n))^2}{f(f(y_n) + y_n) - f(y_n)}$$

Método de Steffensen

Entrada: x, t, n
Salida: x, 'error'

```
leer(x, t, n)
i ← 0
repita
| i ← i + 1
| fx ← f(x)
| y ← x - (fx∗fx / (f(fx + x) - fx))
| epsilon ← |x-y|
| x ← y
hasta (i≥n)∨(epsilon ≤ t)
si (epsilon≤t)
| escribir(x)
sino
| escribir('error')
```

Método iterativo de un punto

Comentarios:

- A medida que $|g'(x)|$ sea más cercana a cero la sucesión $\{x_n\}$ coverge a α más rapidamente.
- Newton-Raphson $g(x_n) = x_n - \frac{f(x_n)}{f'(x_n)}$
- Secante (Método iterativo de dos puntos)

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

- Steffensen: Si se resuelve el problema $x = f(x) + x$ entonces

$$g(x_n) = x_n - \frac{(f(x_n))^2}{f(f(x_n) + x_n) - f(x_n)}$$

Raices de polinomios

Sea $P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \quad , \quad a_n \neq 0$

Teorema fundamental del álgebra: Si $P(x)$ es un polinomio de grado $m \geq 1$ con coeficientes reales o complejos, entonces existe al menos un valor $\alpha \in \mathbb{R}$ o $\mathbb{C}$ tal que $P(\alpha) = 0$

Corolario: Existen un conjunto de constantes únicas $\{\alpha_1, \alpha_2, \cdots \alpha_k\} \in \mathbb{R}$ o $\mathbb{C}$ y un conjunto $\{\beta_1, \beta_2, \cdots \beta_k\} \in \mathbb{N}^+$ tales que

$$P(x) = a_n (x - \alpha_1)^{\beta_1} (x - \alpha_2)^{\beta_2} \cdots (x - \alpha_k)^{\beta_k} \quad \text{y} \quad m = \sum_{i=1}^{k} \beta_i$$

Corolario: Sean $P(x)$ y $Q(x)$ polinomios de grado m. Si existe un conjunto de $n > m$ valores distintos $\{x_1, x_2, \cdots x_n\}$, tales que

$$P(x_i) = Q(x_i) \; \forall \; x_i \quad \text{entonces} \quad P(x) = Q(x) \; \forall \; x$$

Evaluación de polinomios

División sintética o regla de Ruffini: al evaluar

$$P(x) = 3x^3 - 4x^2 - 7x + 1 \ ,$$

ralizaremos 2 multiplicaciones para calcular x^3 y x^2 más 3 multiplicaciones entre los coeficientes y las potencias de x, además realizamos 3 sumas o restas de los términos del polinomio. En general, para un polinomio de grado m se realizarían $(2m-1)$ multiplicaciones y m sumas o restas.

Si escribimos el polinomio de la siguiente forma

$$P(x) = ((3x - 4)\, x - 7)\, x + 1 \ ,$$

Reducimos el número de multiplicaciones a 3, manteniendo el número de sumas y restas en 3.

Evaluación de polinomios

División sintética: En general, si un polinomio de grado m se representa en la forma

$$P(x) = \sum_{i=0}^{m} a_i x^i \ ,$$

se realizarían $(2m-1)$ multiplicaciones y m sumas o restas para evaluarlo. En cambio si se representa como

$$P(x) = (\cdots((a_m x + a_{m-1})\, x + a_{m-2}))\cdots)\, x + a_0 \ ,$$

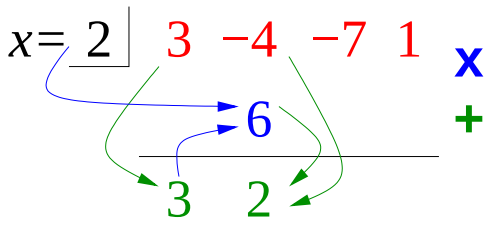
realizaremos solamente m multiplicaciones y m sumas o restas;

¿Cuál es la principal ventaja de reducir el número de operaciones?

Evaluación de polinomios

División sintética Para evaluar el polinomio

$$P(x) = 3x^3 - 4x^2 - 7x + 1 \quad , \text{ en} \quad \bar{x} = 2$$



$x=$ 2 | 3 −4 −7 1

6 4 −6

3 2 −3 | −5 = $P(x)$

2 x 3 = 6

−4 + 6 = 2

2 x 2 = 4

−7 + 4 = −3

2 x −3 = −6

1 + −6 = −5 = $P(x)$

$$\begin{aligned}
b_3 &= a_3 \\
b_2 &= a_2 + a_3\bar{x} \\
&= a_2 + b_3\bar{x} \\
b_1 &= a_1 + (a_2 + a_3\bar{x})\bar{x} \\
&= a_1 + b_2\bar{x} \\
b_0 &= a_0 + \cdots \\
&= a_0 + b_1\bar{x} \\
&= P(\bar{x})
\end{aligned}$$

Evaluación de polinomios

División sintética Para un polinomio de grado m, cuyos coeficientes son $a_m, a_{m-1}, \ldots, a_1, a_0$ tenemos

$$b_m = a_m \quad , \quad b_i = a_i + b_{i+1}\bar{x} \ , \ i = (m-1), (m-2), \ldots, 1, 0 \ ,$$

donde, el último término que se calcula, $b_0 = P(\bar{x})$

El nombre **división sintética** viene del hecho de que los números $b_m, b_{m-1}, \ldots, b_1$ representan los coeficientes de otro polinomio

$$\underbrace{Q(x) = \sum_{i=0}^{m-1} b_{i+1}x^i}_{Cociente} = \frac{\overbrace{P(x)}^{Numerador}}{\underbrace{(x-\bar{x})}_{Denominador}} - \frac{b_0}{\underbrace{x-\bar{x}}_{Resto}}$$

Evaluación de polinomios

División sintética

Entrada: `a, m, x`
Salida: `b`

```
divSintetica(a, m, x, b)
| b[m] ← a[m]
| para i← m-1 hasta 0
| | b[i] ← a[i] + b[i+1] * x
```

Algoritmo de un subprograma que recibe un arreglo `a[0..m]` con los coeficientes del polinomio $P(x)$, su grado `m` y el valor donde se quiere evaluar `x` y retorna el arreglo `b[0..m]` con el resultado de la evaluacion en su primer campo `b[0]`, y los coeficientes de polinomio $Q(x)$ en el resto de sus campos, `b[1]`, `b[2]`, ..., `b[m]`

Método de Horner

De lo anterior tenemos que

$$\frac{P(x)}{x-\bar{x}} = Q(x) + \frac{b_0}{x-\bar{x}}$$

Multiplicando por $(x-\bar{x})$ obtenemos

$$P(x) = Q(x)(x-\bar{x}) + b_0$$

La derivada de lo anterior es

$$P'(x) = Q'(x)(x-\bar{x}) + Q(x)$$

y evaluando la derivada en $x = \bar{x}$ obtenemos

$$P'(\bar{x}) = Q(\bar{x})$$

Método de Horner

Se utiliza para calcular raices de polinomios con el método de Newton-Raphson. Mediante la división sintética se calcula $P'(x) = Q(x)$ y se evalua en un valor dado a $P(x)$ y $Q(x)$

```
Entrada: a, m, x
Salida: b, px, qx
horner(a, m, x, b, px, qx)
| b[m-1] ← a[m]
| qx ← a[m]
| para i ← m-1 hasta 1
| | b[i-1] ← a[i] + b[i] * x
| | qx ← b[i] + qx * x
| px ← a[0] + b[0] * x
```

A medida que se hace la división sintética para evaluar a $P(x)$ en x, se obtienen los coeficientes de $Q(x)$ que son usados dentro la misma iteración para evaluar $Q(x)$ en x.

Al finalizar obtenemos

px $= P($x$)$, qx $= Q($x$) = P'($x$)$

Método de Horner

Si el método obtiene una aproximación a una raiz $P(\alpha_1) \approx 0$, entonces

$$P(x) = (x - \alpha_1)Q_1(x) + P(\alpha_1) \approx (x - \alpha_1)Q_1(x)$$

Este proceso se llama deflación. Si aplicamos la deflación a $Q_1(x)$ tendremos una aproximación a una nueva raiz

$$Q_1(\alpha_2) \approx P(\alpha_2) \approx 0,$$

y si lo repetimos $k = m - 2$ veces se podría tener

$$P(x) \approx (x - \alpha_1)(x - \alpha_2) \cdots (x - \alpha_k)Q_k(x) \ ,$$

donde, $Q_k(x)$ es un polinomio de grado 2, cuyas raices se pueden calcular usando una expresión analítica.

Método de Horner

Entrada: m, a, x, t, n
Salida: Las m raices de $P(x)$

```
deflacion(m, a, x, t, n)
si (m > 2)
| i ← 0
| repita
| | i ← i + 1
| | xa ← x
| | horner(a, m, xa,
| |          b, px, qx)
| | x ← xa - px/qx
| hasta (i≥n)∨(|x-xa|≤t)
| si (|x-xa|≤t)
| | escribir(x)
| | deflacion(m-1, b, x,
| |                t, n)
| sino
| | escribir('error')
sino
| d=sqrt(a[1]∗a[1]
|          - 4∗a[2]∗a[0])
| si(d ≥ 0)
| | escribir((-a[1] ± d)
| |              /(2∗a[2])
```

El subprograma deflacion debe ser llamado por otro programa.

Método de Horner

Comentarios del método:

- Una raiz aproximada α_i, más que una aproximación a una raiz de $P(x)$, es efectivamente una aproximación a una raiz de $Q_{i-1}(x)$ que a su vez se obtiene iterativamente mediante la deflación aproximada de $P(x)$. Esto trae como consecuencia que a medida que i aumenta los errores se acumulan y las aproximaciones α_i a los valores reales de las raices,en general, serán cada vez peores.
- Por lo general los polinomios tiene raices complejas. Para calcularlas utilizando este método (Newton-Raphson) hay que operar usando números complejos.
- Las raices múltiples se van encontrando una a una.

Método de Müller

Teorema: Si $z = \beta + \gamma i$ es una raiz compleja de multiplicidad ν del polinomio $P(x)$ entonces $z = \beta - \gamma i$ también es una raiz de multiplicidad ν de $P(x)$ y el polinomio $D(x) = (x^2 - 2\beta x + \beta^2 + \gamma^2)^\nu$ es un factor de $P(x)$, es decir,

$$\frac{P(x)}{D(x)} = Q(x)$$

Con este resultado podemos pensar en generalizar la división sintética para que los términos sean polinomios cuadráticos. El método de Müller se basa sobre esta idea, pero a diferencia del método de Horner también permite calcular las raices complejas del polinomio.

En cambio de aproximar a la función con una recta, como en el método de la secante, se aproxima usando un polinomio cuadrático

Método de Müller

Sean x_0, x_1 y x_2 los tres puntos iniciales del método y

$$D(x) = a(x - x_2)^2 + b(x - x_2) + c$$

que pasa por $(x_0, P(x_0)), (x_1, P(x_1))$ y $(x_2, P(x_2))$, tenemos que

$$D(x - x_2) = 0 \quad \Rightarrow \quad x - x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Para evitar problemas de errores por la resta de números parecidos y seleccionando el valor de $x_3 = x$ tal que sea la raiz más cercana a x_2

$$x_3 = x_2 - \frac{2c}{b + sig(b)\sqrt{b^2 - 4ac}} \quad , \quad sig(b) = \begin{cases} -1, & \text{si } b < 0 \\ 1, & \text{si } b \geq 0 \end{cases}$$

Método de Müller

Como

$$\begin{aligned} P(x_0) &= a(x_0 - x_2) + b(x_0 - x_2) + c \\ P(x_1) &= a(x_1 - x_2) + b(x_1 - x_2) + c \\ P(x_2) &= a(x_2 - x_2) + b(x_2 - x_2) + c = c \end{aligned}$$

resolviendo el sistema de tres equaciones obtenemos

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

$$b = \frac{P(x_0)-P(x_2)}{(x_0-x_2)} - \underbrace{\frac{(x_0 - x_2)(P(x_1) - P(x_2)) - (x_1 - x_2)(P(x_0) - P(x_2))}{(x_1 - x_2)(x_0 - x_1)}}_{a(x_0-x_2)}$$

$$c = P(x_2)$$

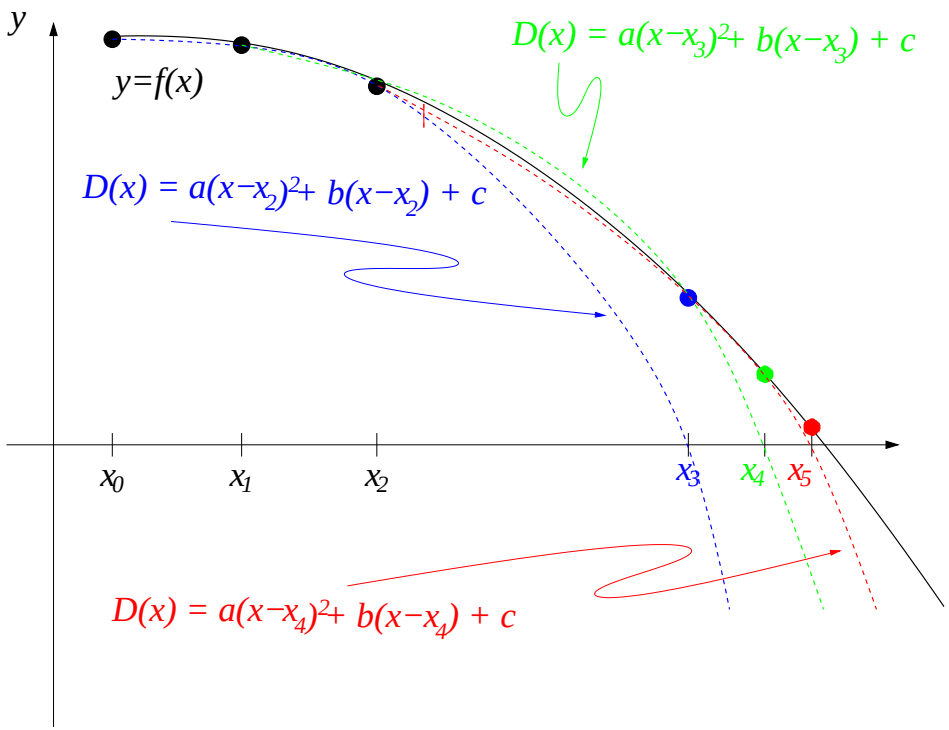
En general podemos tener

$$x_{n+1} = x_n - \frac{2c}{b + sig(b)\sqrt{b^2 - 4ac}}$$

Método de Müller

Entrada: x0,x1,x2,t,n
Salida: x2, 'error'

```
leer(x0,x1,x2,t,n)
repita
| c ← P(x2)
| P02 ← P(x0)-c ; x02 ← x0-x2
| P12 ← P(x1)-c ; x12 ← x1-x2
| a ← (x02∗P12 - x12∗P02)
|       /(x02∗x12∗(x0-x1))
| b ← P02/x02 - a∗x02
| x0 ← x1
| x1 ← x2
| x2 = x2 - 2∗c/(b + sig(b)
|    ∗sqrt(b∗b - 4∗a∗c))
hasta (i≥n)∨(|x2-x1|≤t)
si (|x2-x1|≤t) escribir(x2)
sino escribir('error')
```



Nota: x0, x1 y x2 son variables complejas. En caso de que el lenguaje no las soporte hay que modificar el algoritmo

Algebra Lineal

- Introducción
- Métodos Directos
  - Regla de Crammer
  - Eliminación Gaussiana
  - Factorización LU
  - Factorización de Cholesky
  - Matrices tridiagonales
  - Normas y errores
- Métodos Iterativos
  - Refinamiento iterativo
  - Jacobi
  - Gauss-Seidel
  - Errores y Convergencia
  - Aceleración S.O.R.

Algebra Lineal Numérica

Los problemas más importantes que son atacados con el álgebra lineal númerica son:

- Resolución de sistemas de ecuaciones lineales
- Cálculo de valores propios
- Ajuste de rectas

El problema básico se puede resumir en

$$\mathbf{Ax} = \mathbf{b} \ ,$$

donde, **A** es una matriz $n \times n$, y **x** y **b** son dos vectorer columna $n \times 1$, siendo **A** y **b** conocidos

Regla de Cramer

$$x_i = \frac{\det(\mathbf{A}_i)}{\det(\mathbf{A})} \ ,$$

donde, $\mathbf{A}_i$ es la matriz **A** con la i-esima columna remplazada por **b**

Número de operaciones

- para el cálculo de cada determinante se requieren $(n-1)n!$ multiplicaciones y $(n!-1)$ sumas
- evaluar $(n+1)$ determinantes
- realizar n divisiones

En total son $(n^2+n)n!-1$ operaciones

Eliminación de Gauss

transformar mediante operaciones lineales entre filas a

$$\mathbf{Ax} = \mathbf{b} \quad \text{en} \quad \mathbf{Lx} = \mathbf{c} \quad \text{o en} \quad \mathbf{Ux} = \mathbf{d}\ ,$$

donde,

L

$$\begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ l_{n-11} & l_{n-12} & \cdots & l_{n-1n-1} & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn-1} & l_{nn} \end{bmatrix}$$

Triangular Inferior

U

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n-1} & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n-1} & u_{2n} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & u_{n-1n-1} & u_{n-1n} \\ 0 & \cdots & 0 & 0 & u_{nn} \end{bmatrix}$$

Triangular Superior

Eliminación de Gauss

Sustitución hacia atrás

Triangular Inferior	Triangular Superior
$x_1 = c_1/l_{11}$	$x_n = d_n/u_{nn}$
$x_i = (c_i - \sum_{j=1}^{i-1} l_{ij}x_j)/(l_{ii})$	$x_i = (d_i - \sum_{j=i+1}^{n} u_{ij}x_j)/(u_{ii})$
$i = 2, 3, \ldots, n$	$i = n-1, n-2, \ldots 1$

Para obtener, por ejemplo, la matriz **U** y el vector **d** a partir de **A** y **b** se procede de la siguiente forma sobre la matriz ampliada $\tilde{\mathbf{A}} = \mathbf{A}|\mathbf{b}$

$$a_{ij}^{(1)} = a_{ij}^{(0)} - a_{1j}^{(0)} a_{i1}^{(0)} / a_{11}^{(0)} \ , \quad i = 2, 3, \ldots, n \ , \quad j = 1, 2, \ldots, n+1 \ ;$$

donde, $a_{ij}^{(1)}$ es el elemento que está en la fila i, columna j de $\tilde{\mathbf{A}}$ luego de aplicarle una transformación y $a_{ij}^{(0)}$ es el elemento original.

Eliminación de Gauss

Note que $a_{i1} = 0 \; \forall \; i > 1$.

En este paso a a_{11} se le conoce como el pivote.

Para acomodar la segunda columna de **Ã** usamos a a_{22} como pivote

$$a_{ij}^{(2)} = a_{ij}^{(1)} - a_{2j}^{(1)} a_{i2}^{(1)} / a_{22}^{(1)} \; , \quad i = 3, 4, \ldots, n \; , \quad j = 2, 3, \ldots, n+1 \; ;$$

Repitiendo este proceso para tenemos que

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{kj}^{(k-1)} a_{ik}^{(k-1)} / a_{kk}^{(k-1)} \; ,$$

con

$$k = 1, 2 \ldots, n-1 \; , \quad i = k+1, k+2, \ldots, n \; , \quad j = k, k+1, \ldots, n+1 \; ;$$

donde, $a_{kk}^{(k-1)}$ es el pivote de la k-esima transformación

Eliminación de Gauss

Asi que

$$U = \begin{bmatrix} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn}^{(n-1)} \end{bmatrix} \quad d = \begin{bmatrix} a_{1n+1}^{(0)} \\ a_{2n+1}^{(1)} \\ a_{3n+1}^{(2)} \\ \vdots \\ a_{nn+1}^{(n-1)} \end{bmatrix}$$

Note que para que la eliminación de Gauss funcione todos los $n-1$ pivotes $a_{kk}^{(k-1)}$ tienen que ser distintos de cero. Pero en la práctica esto no es todo, porque pivotes pequeños pueden provocar que el error de redondeo aumente

Eliminación de Gauss

Crecimiento del error por pivotes pequeños[a]

$$\begin{bmatrix} 0.0004 & 1.402 \\ 0.4003 & -1.502 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.406 \\ 2.501 \end{bmatrix}, \quad \frac{0.4003}{0.0004} = 1001$$

Aplicando la eliminación de Gauss tenemos que

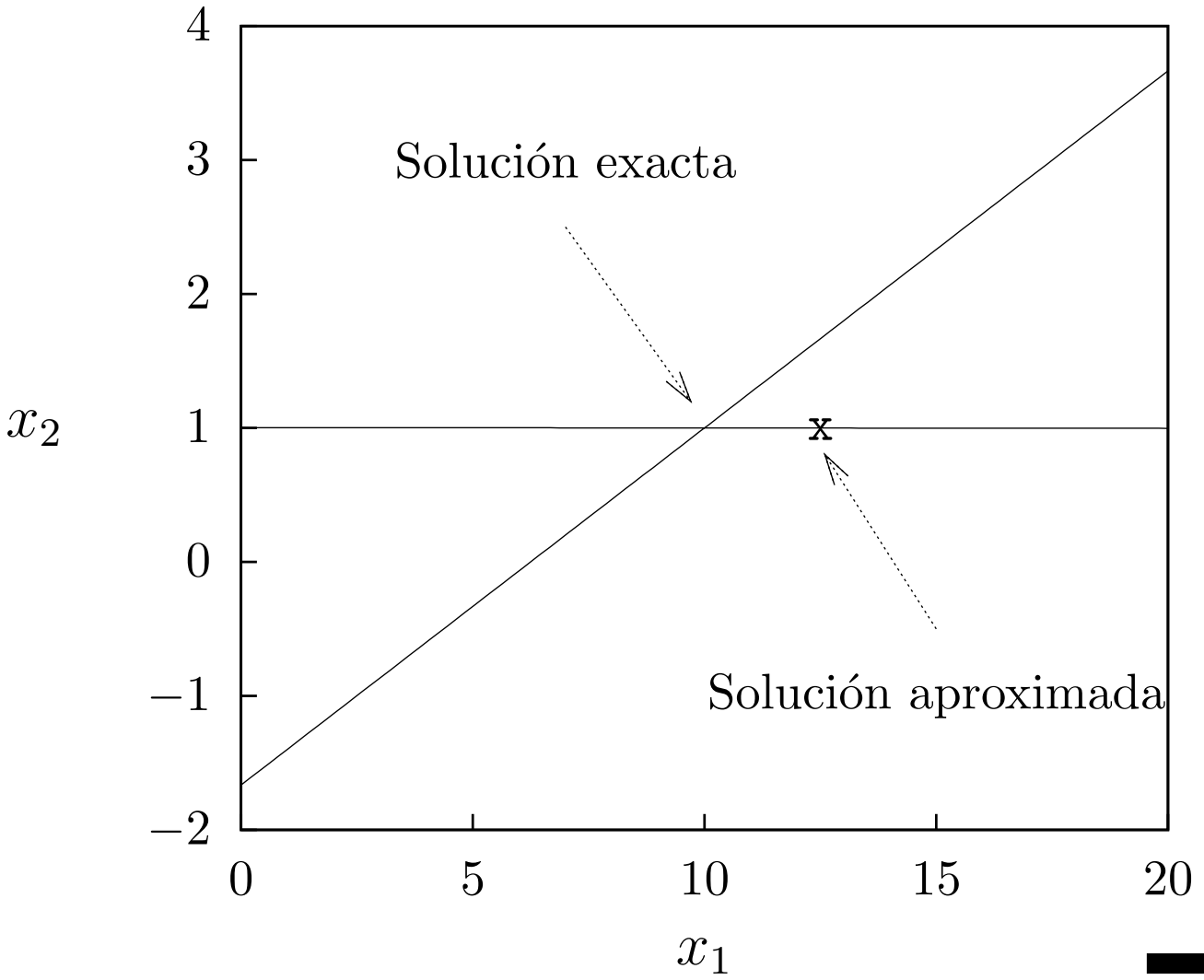
$$\begin{bmatrix} 0.0004 & 1.402 \\ 0 & -1405 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.406 \\ -1404 \end{bmatrix}$$

de donde se obtiene que

$$x_2 = \frac{-1404}{-1405} = 0.9993 \quad \text{y} \quad x_1 = \frac{1.406 - 1.402 \times 0.9993}{0.0004} = 12.5$$

[a] usando aritmética de punto flotante de cuatro cifras decimales

Eliminación de Gauss



Eliminación de Gauss

Crecimiento del error por pivotes pequeños[a]

Si ahora usamos la eliminación de Gauss con una matriz **L** tenemos

$$\begin{bmatrix} 0.0004 & 1.402 \\ 0.4003 & -1.502 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.406 \\ 2.501 \end{bmatrix}, \quad \frac{1.402}{-1.502} = -0.9334$$

Aplicando la eliminación de Gauss tenemos que

$$\begin{bmatrix} 0.374 & 0 \\ 0.4003 & -1.502 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3.74 \\ 2.501 \end{bmatrix}$$

de donde se obtiene que

$$x_1 = \frac{3.74}{0.374} = 10 \quad \text{y} \quad x_1 = \frac{2.501 - 0.4003 \times 10}{-1.502} = 1$$

[a] usando aritmética de punto flotante de cuatro cifras decimales

Eliminación de Gauss

Estrategias de pivoteo

- **Pivoteo Parcial:** Se escoje r tal que

$$\left|a_{rk}^{(k-1)}\right| = \max_{k \leq i \leq n} \left(\left|a_{ik}^{(k-1)}\right|\right)$$

luego se intercambian la fila k con la fila r

Menos efectivo, menos costoso, más usado

- **Pivoteo Total:** Se escoje r y s tales que

$$\left|a_{rs}^{(k-1)}\right| = \max_{k \leq i,j \leq n} \left(\left|a_{ij}^{(k-1)}\right|\right)$$

Se intercambian la fila k con la r y la columna k con la s

Más efectivo, más costoso, menos usado

Eliminación de Gauss

```
Entrada: n, A[n,n+1]
Salida: Las x[n]
gauss(n, A, x)
para k ← 1 hasta n
| f[k] ← k
para k ← 1 hasta n-1
| r ← k
| para i ← k+1 hasta n
| | si (A[i,k] > A[r,k]
| | | r ← i
| | si A[r,k] = 0
| | | escribir("Singular")
| | | retornar
| | si (k ≠ r)
| | | aux ← f[k]
| | | f[k] ← f[r]
| | | f[r] ← aux
| | m ← A[f[i],k]/A[f[k],k]
| | para j ← k hasta n+1
| | | A[f[i],j] ← A[f[i],j]
| | |       - m*A[f[k],j]
si A[f[n],n] = 0
| escribir("Singular")
x[n] ← A[f[n],n+1]/Af[n],n]
para i ← n-1 hasta 1
| x[i] ← 0
| para j ← i+1 hasta n
| | x[i] ← x[i]
| |      + A[f[i],j]*x[j]
| x[i] ← (A[f[i],n+1]
|        - x[i])/A[f[i],i]
retornar
```

Eliminación de Gauss

Comentarios

- El número de operaciones es aproximadamente $\frac{2}{3}n^2$
- Si se quieren resolver varios sistemas con la misma matriz **A** la matriz ampliada será $\tilde{\mathbf{A}} = \mathbf{A}|\mathbf{b}_1|\mathbf{b}_2|\cdots$
- Para el cálculo de la inversa de la matriz **A** aplicamos el método de la eliminación de Gauss a la matriz ampliada $\tilde{\mathbf{A}} = \mathbf{A}|\mathbf{I}$, donde **I** es la matriz identidad
- El determinante de **A** se puede calcular a partir de la diagonal de la matriz triangular

$$\det(\mathbf{A}) = (-1)^s \prod_{k=1}^{n} a_{kk}^{(k-1)} \ ,$$

donde s es el número de intercambios de filas realizados por las estrategia de pivoteo parcial

Factorización LU

Si la matriz **A** se usa varias veces es conveniente factorizarla usando una matriz triangular inferior **L** y una triangular superior **U**

$$\mathbf{A} = \mathbf{LU} \quad ; \quad \mathbf{Ax} = \mathbf{b} \quad \Rightarrow \quad \mathbf{Ly} = \mathbf{b} \ , \ \mathbf{Ux} = \mathbf{y}$$

Donde, conocidas **L** y **U** se requieren n^2 operaciones para resolver el sistema $\mathbf{Ax} = \mathbf{b}$

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots & \\ l_{n-11} & l_{n-12} & \cdots & l_{n-1n-2} & 1 & 0 \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn-1} & 1 \end{bmatrix}, \quad l_{ik} = \begin{cases} \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} & \text{, si } i \geq k \\ 0 & \text{, si } i < k \end{cases}$$

Factorización LU

$$\mathbf{U} = \begin{bmatrix} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn}^{(n-1)} \end{bmatrix}, \quad a_{ij}^{(k)} = a_{ij}^{(k-1)} - l_{ik} a_{kj}^{(k-1)}$$

$$k = 1, 2, \ldots, n-1$$

$$i, j = k+1, k+2 \ldots n$$

Teorema: La factorización **LU** de una matriz **A** no singular en única.

Corolario: $\det(\mathbf{A}) = \det(\mathbf{LU}) = \det(\mathbf{L}) \det(\mathbf{U}) = \prod_{k=1}^{n} a_{kk}^{(k-1)}$

Si se utilizan estrategias de pivoteo parcial o total, la factorización **LU** será de la matriz permutada **PA** donde la matriz **P**, llamada matriz de permutación, es el resultado de realizar las permutaciones hechas en **A** en la matriz identidad **I**

Método de Cholesky

Si la matriz **A** es simétrica y definida positiva no es necesario aplicar estrategias de pivoteo

Teorema de Cholesky: Sea **A** una matriz simétrica definida positiva. Entonces existe una única matriz triangular inferior **L** con $l_{ii} \neq 0$ tal que

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T$$

- **A** es simétrica si:

$$\mathbf{A} = \mathbf{A}^T \quad , \quad a_{ij} = a_{ji} \; \forall \; i, j$$

  - $\mathbf{A}^{-1}$, si existe es simétrica
  - Las submatrices principales de **A** son simétricas

Método de Cholesky

- **A** es definida positiva si:

$$\mathbf{x}^T\mathbf{A}\mathbf{x} > 0 \quad \forall \quad \mathbf{x} \neq 0$$

  - $a_{ii} > 0 \; \forall \; i$
  - $\mathbf{A}^T$ es definida positiva
  - $\mathbf{A}^{-1}$ siempre existe
  - $\mathbf{A}^{-1}$ es definida positiva
  - Las submatrices principales son definidas positivas
  - $\det(\mathbf{A}) > 0$
  - Los menores principales son positivos

- Una matriz simétrica es definida positiva si y solo si todos sus menores principales son positivos *(Teorema de Sylvester)*

Para la demostración del teorema de Cholesky ver las notas del curso "Análisis Numérico", de la Prof. María C. Trevisan.

http://www.matematicas.ula.ve/publicaciones/guias/cursonum.pdf

Método de Cholesky

- El método requiere $n^3/3$ operaciones, la mitad de las $2n^3/3$ operaciones requeridas para la factorización **LU**
- Se deben calcular n raices cuadradas
- Los elementos l_{ii} de **L** pueden ser diferente de 1
- Si se quiere eliminar el cálculo de las n raices cuadradas se puede modificar la factorización a

$$\mathbf{A} = \mathbf{LDL}^T \text{ , donde,}$$

$$l_{ii} = 1 \quad , \quad l_{i1} = a_{i1} \quad \text{y} \quad l_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} l_{jk}^2 d_{kk} \right) / d_{ii}$$

y **D** es una matriz diagonal con

$$d_{11} = a_{11} \quad \text{y} \quad d_{ii} = a_{ii} - \sum_{j=1}^{i-1} l_{ij}^2 d_{jj}$$

Método de Cholesky

```
Entrada: n, A[n,n],b[n]
Salida: Las x[n]
cholesky(n, A, b x)
para i ← 1 hasta n
| para j ← 1 hasta i-1
| | v[j] = l[i,j]*d[j]
| d[i] ← A[i,i]
| para j ← 1 hasta i-1
| | d[i] ← d[i]
| |       - l[i,j]*v[j]
| para j ← i+1 hasta n
| | l[i,j] ← A[j,i]
| | para k ← 1 hasta i-1
| | | l[i,j] ← l[i,j]
| | |       - l[j,k]*v[k]
| | l[i,j] ← l[j,k]/d[i]
```

```
para i ← 1 hasta n
| y[i] ← b[i]
| para j ← 1 hasta i-1
| | y[i] ← y[i]
| |       - l[i,j]*y[j]
para i ← 1 hasta n
| z[i] ← y[i]/d[i]
para i ← n hasta 1
| x[i] ← z[i]
| para j ← i+1 hasta n
| | x[i] ← x[i]
| |       - l[j,i]*x[j]
retornar
```

Matrices Tridiagonales

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & 0 & \cdots & 0 \\ 0 & a_{32} & a_{33} & a_{34} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & & 0 & a_{n-1n-2} & a_{n-1n-1} & a_{n-1n} \\ 0 & \cdots & 0 & 0 & a_{nn-1} & a_{nn} \end{bmatrix}$$

Si la matriz **A** es no singular entonce podemos factorizarla

$$\mathbf{A} = \mathbf{LU} \, ,$$

donde,

Matrices Tridiagonales

L

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ l_1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & l_2 & 1 & \ddots & \vdots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 & 0 \\ 0 & \cdots & 0 & 0 & l_{n-1} & 1 \end{bmatrix}$$

U

$$\begin{bmatrix} u_1 & v_1 & 0 & 0 & \cdots & 0 \\ 0 & u_2 & v_2 & 0 & \ddots & \vdots \\ 0 & 0 & u_3 & v_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & u_{n-1} & v_{n-1} \\ 0 & 0 & \cdots & 0 & 0 & u_n \end{bmatrix}$$

donde,

$$v_i = a_{ii+1}\ , \quad u_1 = a_{11}\ , \quad l_i = \frac{a_{i+1i}}{u_{i-1}}\ , \quad u_i = a_{ii} - l_i v_{i-1}$$

Normas Vectoriales y Matriciales

Definición: Una norma vectorial en $\mathbb{R}^n$ es una función, $\|\cdot\|$, de $\mathbb{R}^n$ en $\mathbb{R}$ con las siguientes propiedades:

- $\| \mathbf{x} \| \geq 0 \quad \forall \quad \mathbf{x} \in \mathbb{R}^n$
- $\| \mathbf{x} \| = 0$ si y sólo si $\mathbf{x} = 0$
- $\| a\mathbf{x} \| = |a| \, \| \mathbf{x} \| \quad \forall \quad a \in \mathbb{R}$ y $\mathbf{x} \in \mathbb{R}^n$
- $\| \mathbf{x} + \mathbf{y} \| \leq \| \mathbf{x} \| + \| \mathbf{y} \| \quad \forall \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

Definición: Sea $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ un vector en $\mathbb{R}^n$, las normas $\| \mathbf{x} \|_p$ y $\| \mathbf{x} \|_\infty$ vienen dadas por

$$\| \mathbf{x} \|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \qquad \text{y} \qquad \| \mathbf{x} \|_\infty = \max_{1 \leq i \leq n} |x_i|$$

La norma $\| \mathbf{x} \|_2$ se conoce como norma Euclidiana
La norma $\| \mathbf{x} \|_\infty$ se conoce como norma uniforme

Normas Vectoriales y Matriciales

Definición: Sean $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ y $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$ dos vectores en $\mathbb{R}^n$, las distancias l_p y l_∞ entre **x** y **y** vienen dadas por

$$\| \mathbf{x} - \mathbf{y} \|_p = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \qquad \text{y} \qquad \| \mathbf{x} - \mathbf{y} \|_\infty = \max_{1 \leq i \leq n} |x_i - y_i|$$

Ejemplo 1: Si $\mathbf{x} = (1, 1, 1)^T$ y $\mathbf{y} = (1.1, 0.99, 1.04)^T$

$$\| \mathbf{x} - \mathbf{y} \|_2 = 0.1081665 \quad , \quad \| \mathbf{x} - \mathbf{y} \|_\infty = 0.1$$

Ejemplo 2: Si $\mathbf{x} = (1, 1, 1)^T$ y $\mathbf{y} = (1.1, 0.999999, 1.0000001)^T$

$$\| \mathbf{x} - \mathbf{y} \|_2 \approx 0.1 \quad , \quad \| \mathbf{x} - \mathbf{y} \|_\infty = 0.1$$

Normas Vectoriales y Matriciales

Definición: Se dice que una sucesión $\{\mathbf{x}^{(k)}\}_{k=1}^{\infty}$ converge a **x** respecto a la norma $\|\cdot\|$ si dado cualquier $\epsilon > 0$, existe un entero $N(\epsilon)$ tal que

$$\| \mathbf{x}^{(k)} - \mathbf{x} \| < \epsilon \quad , \quad \forall \quad k \geq N(\epsilon) \; .$$

Teorema: La sucesión de vectores $\{\mathbf{x}^{(k)}\}$ converge a **x** respecto a la norma $\|\cdot\|_\infty$ si y sólo si

$$\lim_{k \to \infty} x_i^{(k)} = x_i \quad , \quad \forall \quad i = 1, 2, \ldots, n$$

Teorema: Para todo $\mathbf{x} \in \mathbb{R}^n$, $\| \mathbf{x} \|_\infty \leq \| \mathbf{x} \|_2 \leq \sqrt{n} \, \| \mathbf{x} \|_\infty$.
Demostración: Sea x_j la coordenada de **x** tal que $\| \mathbf{x} \|_\infty = |x_j|$. Entonces

$$\| \mathbf{x} \|_\infty^2 = x_j^2 \quad \leq \quad \sum_{i=1}^{n} x_i^2 = \| \mathbf{x} \|_2^2 \quad \leq \quad \sum_{i=1}^{n} x_j^2 = n x_j^2 = n \, \| \mathbf{x} \|_\infty^2$$

Normas Vectoriales y Matriciales

Definición: Una norma matricial en $\mathbb{R}^{n,n}$ es una función, $\|\cdot\|$, de $\mathbb{R}^{n,n}$ en $\mathbb{R}$ tal que:

- $\| \mathbf{A} \| \geq 0 \quad \forall \quad \mathbf{A} \in \mathbb{R}^{n,n}$
- $\| \mathbf{A} \| = 0$ si y sólo si $\mathbf{A} = 0$
- $\| a\mathbf{A} \| = |a| \, \| \mathbf{A} \| \quad \forall \quad a \in \mathbb{R}$ y $\mathbf{A} \in \mathbb{R}^{n,n}$
- $\| \mathbf{A} + \mathbf{B} \| \leq \| \mathbf{A} \| + \| \mathbf{B} \| \quad \forall \quad \mathbf{A}, \mathbf{B} \in \mathbb{R}^{n,n}$
- $\| \mathbf{AB} \| \leq \| \mathbf{A} \| \| \mathbf{B} \| \quad \forall \quad \mathbf{A}, \mathbf{B} \in \mathbb{R}^{n,n}$

Definición: Dada una norma vectorial en $\mathbb{R}^n$, se define la norma matricial inducida por[a] dicha norma vectorial como

$$\| \mathbf{A} \| = \max_{\mathbf{x} \neq 0} \frac{\| \mathbf{Ax} \|}{\| \mathbf{x} \|} = \max_{\|\mathbf{x}\|=1} \| \mathbf{Ax} \|$$

[a] También se le llama subordinada a la norma vectorial

Normas Vectoriales y Matriciales

Teorema: Si $\mathbf{A} = (a_{ij})$ es una matriz $n \times n$, entonces

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

Teorema: Si $\mathbf{A} = (a_{ij})$ es una matriz $n \times n$, entonces

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

Teorema: Si $\mathbf{A} = (a_{ij})$ es una matriz $n \times n$, entonces

$$\| \mathbf{A} \|_2 = \sqrt{(\rho(\mathbf{A}^T \mathbf{A})}\ ,$$

donde, $\rho(\mathbf{A})$ es el radio espectral de $\mathbf{A}$ y se define como

$$\rho(\mathbf{A}) = \max_{\lambda \in \sigma(\mathbf{A})} |\lambda| \quad \text{y} \quad \sigma(\mathbf{A}) \text{ es el espectro de autovalores de } \mathbf{A}$$

Errores

Por lo general, la solución calculada numéricamente

$$\tilde{\mathbf{x}} \approx \mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad \Rightarrow \quad \mathbf{A}\tilde{\mathbf{x}} = \mathbf{b} - \mathbf{r} \quad ,$$

donde **r** es el vector residual. Se puede pensar erroneamente que un vector residual pequeño está asociado a un error pequeño, por ejemplo, si la solución numérica el sistema

$$\begin{array}{rcrl} 0.9999x_1 & - & 1.0001x_2 & = 1 \\ x_1 & - & x_2 & = 1 \end{array} \quad ,$$

es $\tilde{\mathbf{x}} = (1\ ,\ 0)^T$, el vector residual $\mathbf{r} = (0.0001\ ,\ 0)^T$ es pequeño y a pesar de esto el error, que viene dado por la expresión

$$\tilde{\mathbf{x}} - \mathbf{x} = \tilde{\mathbf{x}} - \mathbf{A}^{-1}\mathbf{b} \quad ,$$

es grande, ya que la solución exacta es $\mathbf{x} = (0.5\ ,\ -0.5)^T$

Errores

Error Absoluto

Teorema: Si $\tilde{\mathbf{x}}$ es una aproximación de la solución de $\mathbf{Ax} = \mathbf{b}$ y $\mathbf{A}$ es una matriz no singular, entonces para cualquier norma subordinada el error absoluto

$$\| \mathbf{x} - \tilde{\mathbf{x}} \| \leq \| \mathbf{r} \| \| \mathbf{A} \| \quad ,$$

Demostración: De la definición del vector residual tenemos que

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}} = \mathbf{Ax} - \mathbf{A}\tilde{\mathbf{x}} = \mathbf{A}(\mathbf{x} - \tilde{\mathbf{x}}) \quad \Rightarrow \quad \mathbf{x} - \tilde{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{r}$$

de donde se obtiene

$$\| \mathbf{x} - \tilde{\mathbf{x}} \| = \| \mathbf{A}^{-1}\mathbf{r} \| \leq \| \mathbf{A}^{-1} \| \| \mathbf{r} \|$$

Errores

Error Relativo

Teorema: Si $\tilde{\mathbf{x}}$ es una aproximación de la solución de $\mathbf{Ax} = \mathbf{b}$ y $\mathbf{A}$ es una matriz no singular, entonces para cualquier norma subordinada el error relativo

$$\frac{\| \mathbf{x} - \tilde{\mathbf{x}} \|}{\| \mathbf{x} \|} \leq \frac{\| \mathbf{r} \|}{\| \mathbf{b} \|} \| \mathbf{A} \| \| \mathbf{A}^{-1} \| \quad , \text{si } \mathbf{x} \neq 0, \mathbf{b} \neq 0$$

Demostración: Como

$$\mathbf{b} = \mathbf{Ax} \quad \Rightarrow \quad \| \mathbf{b} \| \leq \| \mathbf{A} \| \| \mathbf{x} \| \quad \Rightarrow \quad \| \mathbf{x} \| \geq \frac{\| \mathbf{b} \|}{\| \mathbf{A} \|}$$

Dividiendo al error absoluto por $\| \mathbf{x} \|$ y a su cota por $\| \mathbf{b} \| / \| \mathbf{A} \|$, tenemos que

$$\frac{\| \mathbf{x} - \tilde{\mathbf{x}} \|}{\| \mathbf{x} \|} \leq \frac{\| \mathbf{r} \|}{\| \mathbf{b} \|} \| \mathbf{A} \| \| \mathbf{A}^{-1} \|$$

Errores

Número de condición

Definición: El número de condición de una matriz no singular **A** relativo a la norma $\|\cdot\|$ es

$$\mathcal{K}(\mathbf{A}) = \| \mathbf{A} \| \| \mathbf{A}^{-1} \|$$

Con esta definición los errores absoluto y relativo se pueden escribir como

$$\| \mathbf{x} - \tilde{\mathbf{x}} \| \leq \mathcal{K}(\mathbf{A}) \frac{\| \mathbf{r} \|}{\| \mathbf{A} \|} \quad , \quad \frac{\| \mathbf{x} - \tilde{\mathbf{x}} \|}{\| \mathbf{x} \|} \leq \frac{\| \mathbf{r} \|}{\| \mathbf{b} \|} \| \mathbf{A} \| \| \mathbf{A}^{-1} \|$$

Una matriz **A** está *bien condicionada* si $\mathcal{K}(\mathbf{A})$ esta cerca de 1 y está *mal condicionada* si $\mathcal{K}(\mathbf{A}) \gg 1$

Un buen condicionamiento garantiza que un vector residual pequeño implica que el error de la solución aproximada es pequeño.

Calcule el número de condición para la matriz del problema anterior

Metodos Iterativos

Para resolver el problema

$$\mathbf{Ax} = \mathbf{b} \quad ,$$

un método iterativo comienza con una aproximación inicial de la solución $\mathbf{x}^{(0)}$, y genera una sucesión de vectores $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ que converge a la solución $\mathbf{x}$; convirtiendo el problema en uno equivalende de la forma

$$\mathbf{x} = \mathbf{Bx} + \mathbf{c} \quad ,$$

donde, $\mathbf{B}$ es una matriz constante $n \times n$ y $\mathbf{c}$ es un vector columna también constante.
Si una sucesión de vectores $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ tiende a $\mathbf{x}$, es decir que

$$\lim_{k \to \infty} \| \mathbf{x}^{(k)} - \mathbf{x} \| = 0 \quad ,$$

entonces, $\mathbf{x}^{(k \to \infty)}$ es solución de

$$\mathbf{x} = \mathbf{Bx} + \mathbf{c} \quad \text{y} \quad \mathbf{Ax} = \mathbf{b}$$

Métodos Iterativos

Veremos que para generalizar, es conveniente expresar a la matriz **A** como

$$\mathbf{A} = \mathbf{D}(\mathbf{L} + \mathbf{I} + \mathbf{U}) \quad ,$$

donde,**I** en la matriz identidad;

$$d_{ij} = \begin{cases} a_{ii} & , \text{si } j = i \\ & \\ 0 & , \text{si } j \neq i \end{cases} ;$$

$$l_{ij} = \begin{cases} a_{ij}/a_{ii} & , \text{si } j < i \\ & \\ 0 & , \text{si } j \geq i \end{cases} ; \quad u_{ij} = \begin{cases} a_{ij}/a_{ii} & , \text{si } j > i \\ & \\ 0 & , \text{si } j \leq i \end{cases}$$

Métodos Iterativos

Despejamos por cada una de las equaciones una variables diferente, por ejemplo

$$\begin{array}{rcl} -4x_1 + 3x_2 + x_3 &=& 1 \\ x_1 + 5x_2 - x_3 &=& 2 \\ 2x_1 + 3x_2 + 3x_3 &=& 4 \end{array} \quad \Rightarrow \quad \begin{array}{rcl} x_1 &=& \frac{3}{4}x_2 + \frac{1}{4}x_3 - \frac{1}{4} \\ x_2 &=& -\frac{1}{5}x_1 + \frac{1}{5}x_3 + \frac{2}{5} \\ x_3 &=& -\frac{2}{3}x_1 - x_2 + \frac{4}{3} \end{array}$$

En general podemos decir que

$$x_i = \left(b_i - \sum_{j \neq i} a_{ij} x_j \right) / a_{ii} \quad , \quad i = 1, 2 \ldots, n$$

Ordenando las equaciones de tal forma que todos los $a_{ii} \neq 0$

Método de Jacobi

Para calcular la sucesión de vectores $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ comenzamos con un vector inicial $\mathbf{x}^{(0)}$ y usando la expresión anterior calculamos

$$x_i^{(k+1)} = \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right) / a_{ii} \quad , \qquad \begin{array}{l} i = 1, 2, \ldots, n \\ k = 0, 1, \ldots \end{array}$$

Escrito en forma vectorial tenemos

$$\mathbf{x}^{(k+1)} = -(\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b} \quad ,$$

es decir que en la expresión iterativa

$$\mathbf{x}^{(k+1)} = \mathbf{B}\mathbf{x}^{(k)} + \mathbf{c} \quad , k = 0, 1, \ldots$$

la matriz de iteración $\mathbf{B}$ y el vector $\mathbf{c}$ vienen dados por

$$\mathbf{B}_J = -(\mathbf{L} + \mathbf{U}) \quad , \quad \mathbf{c}_J = \mathbf{D}^{-1}\mathbf{b}$$

Método de Jacobi

```
Entrada: n,A[n,n],b[n],x[n],t,m
Salida: x[n], e
jacobi(n, A, b, t, m, x, e)
k ← 0
repita
| k ← k + 1
| para i ← 1 hasta n
| | xa[i] ← x[i]
| para i ← 1 hasta n
| | x[i] ← b[i]
| |      + A[i,i]∗xa[i]
| | para j ← 1 hasta n
| | | x[i] ← x[i]
| | |      - A[i,j]∗xa[j]
| | x[i] ← x[i]/A[i,i]
| para i ← 1 hasta n
| | d[i] ← x[i] - xa[i]
| si(norma(n,d)> tol)
| | e ← verdadero
| sino
| | e ← falso
hasta((¬ e) ∨ (k>m))
retornar
```

Note que para la condición de parada estamos usando la función norma que calcula la norma vectorial de e

Método de Gauss-Seidel

Para calcular la sucesión de vectores $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ comenzamos con un vector inicial $\mathbf{x}^{(0)}$, pero a diferencia del método de Jacobi donde para calcular el valor de $x_i^{(k+1)}$ se usan solamente los valores estimados en la iteración anterior, en este método para calcular el valor de $x_i^{(k+1)}$ usamos los $x_j, j = 1, 2, \ldots i-1$ valores ya calculados de la iteración actual y $x_j, j = i+1, i+2, \ldots n$ valores de la iteración anterior.

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) / a_{ii} \quad , \qquad \begin{array}{l} i = 1, 2, \ldots, n \\ k = 0, 1, \ldots \end{array}$$

Escrito en forma vectorial tenemos

$$\mathbf{x}^{(k+1)} = -\mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{U}\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b} \quad ,$$

Método de Gauss-Seidel

de donde

$$(\mathbf{I}+\mathbf{L})\mathbf{x}^{(k+1)} = -\mathbf{U}\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b} \quad ,$$

y por lo tanto

$$\mathbf{x}^{(k+1)} = -(\mathbf{I}+\mathbf{L})^{-1}\mathbf{U}\mathbf{x}^{(k)} + (\mathbf{I}+\mathbf{L})^{-1}\mathbf{D}^{-1}\mathbf{b} \quad ,$$

es decir que en la expresión iterativa

$$\mathbf{x}^{(k+1)} = \mathbf{B}\mathbf{x}^{(k)} + \mathbf{c} \quad , k = 0, 1, \ldots$$

la matriz de iteración **B** y el vector **c** vienen dados por

$$\mathbf{B}_{GS} = -(\mathbf{I}+\mathbf{L})^{-1}\mathbf{U} \quad , \quad \mathbf{c}_{GS} = (\mathbf{I}+\mathbf{L})^{-1}\mathbf{D}^{-1}\mathbf{b}$$

Método de Gauss-Seidel

```
Entrada: n,A[n,n],b[n],x[n],t,m
Salida: x[n], e
gs(n, A, b, t, m, x, e)
k ← 0
repita
| k ← k + 1
| para i ← 1 hasta n
| | xa[i] ← x[i]
| para i ← 1 hasta n
| | x[i] ← b[i]
| | para j ← 1 hasta i-1
| | | x[i] ← x[i]
| | |      - A[i,j]*x[j]
| | para j ← i+1 hasta n
| | | x[i] ← x[i]
| | |      - A[i,j]*xa[j]
| | x[i] ← x[i]/A[i,i]
| para i ← 1 hasta n
| | d[i] ← x[i] - xa[i]
| si(norma(n,d)> tol)
| | e ← verdadero
| sino
| | e ← falso
hasta((¬ e) ∨ (k>m))
retornar
```

Note que para la condición de parada estamos usando la función norma que calcula la norma vectorial de d

Métodos Iterativos

Unicidad

La ecuación vectorial $\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{c}$ tiene una única solución ya que $(\mathbf{I} - \mathbf{B})^{-1}$ existe.
Para Jacobi viene expresada como

$$\mathbf{I} - \mathbf{B}_J = \mathbf{L} + \mathbf{I} + \mathbf{U} = \mathbf{D}^{-1}\mathbf{A}$$

y para Gauss-Seidel como

$$\begin{aligned} \mathbf{I} - \mathbf{B}_{GS} &= \mathbf{I} + (\mathbf{I} + \mathbf{L})^{-1}\mathbf{U} = (\mathbf{I} + \mathbf{L})^{-1}(\mathbf{I} + \mathbf{L}) + (\mathbf{I} + \mathbf{L})^{-1}\mathbf{U} \\ &= (\mathbf{I} + \mathbf{L})^{-1}(\mathbf{L} + \mathbf{I} + \mathbf{U}) = (\mathbf{I} + \mathbf{L})^{-1}\mathbf{D}^{-1}\mathbf{A} \end{aligned}$$

Convergencia

Lema 1: Si el radio espectral $\rho(\mathbf{B}) < 1$ entonces $(\mathbf{I} - \mathbf{B})^{-1}$ existe y además

$$(\mathbf{I} - \mathbf{B})^{-1} = \mathbf{I} + \mathbf{B} + \mathbf{B}^2 + \cdots = \sum_{k=0}^{\infty} \mathbf{B}^k$$

Métodos Iterativos

Convergencia

Lema 2: Las siguientes afirmaciones son equivalentes

1. $\lim_{k\to\infty} a_{ij}^k = 0$
2. $\lim_{k\to\infty} \| \mathbf{A}^k \| = 0$
3. $\rho(\mathbf{A}) < 1$
4. $\lim_{k\to\infty} \mathbf{A}^k\mathbf{x} = 0 \quad \forall \quad \mathbf{x}$

Lema 3: $\rho(\mathbf{A}) < \| \mathbf{A} \|$, para cualquier norma inducida
Prueba: Sea λ tal que $\rho(\mathbf{A}) = |\lambda|$ y **u** su autovector asociado. Entonces

$$\mathbf{Au} = \lambda\mathbf{u} \Rightarrow \| \mathbf{Au} \| = |\lambda| \, \| \mathbf{u} \|$$

$$\Rightarrow \rho(\mathbf{A}) = |\lambda| = \frac{\| \mathbf{Au} \|}{\| \mathbf{u} \|} \leq \max_{x\neq 0} \frac{\| \mathbf{Ax} \|}{\| \mathbf{x} \|} = \| \mathbf{A} \|$$

Métodos Iterativos

Convergencia

Teorema: Para cualquier $\mathbf{x}^{(0)} \in \mathbb{R}^n$, la sucesión $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ determinada por

$$\mathbf{x}^{(k+1)} = \mathbf{B}\mathbf{x}^{(k)} + \mathbf{c} \quad , \quad k = 0, 1, \ldots \quad \text{y} \quad \mathbf{c} \neq 0$$

converge a una solución única, $\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{c}$, si y sólo si $\rho(\mathbf{B}) < 1$

Demostración

$$\begin{aligned}
\mathbf{x}^{(1)} &= \mathbf{B}\mathbf{x}^{(0)} + \mathbf{c} \\
\mathbf{x}^{(2)} &= \mathbf{B}\mathbf{x}^{(1)} + \mathbf{c} = \mathbf{B}^2\mathbf{x}^{(0)} + (\mathbf{B} + \mathbf{I})\mathbf{c} \\
\mathbf{x}^{(3)} &= \mathbf{B}\mathbf{x}^{(2)} + \mathbf{c} = \mathbf{B}^3\mathbf{x}^{(0)} + (\mathbf{B}^2 + \mathbf{B} + \mathbf{I})\mathbf{c} \\
&\vdots \quad \vdots \quad \vdots \\
\mathbf{x}^{(k)} &= \mathbf{B}\mathbf{x}^{(k-1)} + \mathbf{c} = \mathbf{B}^k\mathbf{x}^{(0)} + (\mathbf{B}^{k-1} + \cdots + \mathbf{B} + \mathbf{I})\mathbf{c} \\
&= \mathbf{B}^k\mathbf{x}^{(0)} + \left(\sum_{j=0}^{k-1} \mathbf{B}^j\right)\mathbf{c}
\end{aligned}$$

Métodos Iterativos

Convergencia

Si $\rho(\mathbf{B}) < 1$,

$$\lim_{k\to\infty} \mathbf{x}^{(k)} = \underbrace{\lim_{k\to\infty} \mathbf{B}^k \mathbf{x}^{(0)}}_{lema2} + \underbrace{\lim_{k\to\infty} \left(\sum_{j=0}^{k-1} \mathbf{B}^j \right) \mathbf{c}}_{lema1} \quad ,$$

entonces,

$$\lim_{k\to\infty} \mathbf{x}^{(k)} = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{c}$$

es decir, la sucesión $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ converge si $\rho(\mathbf{B}) < 1$

Métodos Iterativos

Convergencia

Por otro lado, sea $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ tal que

$$\lim_{k\to\infty} \mathbf{x}^{(k)} = \mathbf{x}$$

entonces, $\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{c}$ y $\mathbf{x}^{(k)} = \mathbf{B}\mathbf{x}^{(k-1)} + \mathbf{c}$, y el error viene dado por

$$\begin{aligned}
\mathbf{x} - \mathbf{x}^{(k)} &= \mathbf{B}\mathbf{x} + \mathbf{c} - \mathbf{B}\mathbf{x}^{(k-1)} - \mathbf{c} = \mathbf{B}\mathbf{x} - \mathbf{B}\mathbf{x}^{(k-1)} = \mathbf{B}(\mathbf{x} - \mathbf{x}^{(k-1)}) \\
&= \mathbf{B}(\mathbf{B}\mathbf{x} + \mathbf{c} - \mathbf{B}\mathbf{x}^{(k-2)} - \mathbf{c}) = \mathbf{B}^2(\mathbf{x} - \mathbf{x}^{(k-2)}) \\
&= \cdots \\
&= \mathbf{B}^k(\mathbf{x} - \mathbf{x}^{(0)}) \quad \text{Por lo que}
\end{aligned}$$

$$\lim_{k\to\infty} \mathbf{x} - \mathbf{x}^{(k)} = \lim_{k\to\infty} \mathbf{B}^k(\mathbf{x} - \mathbf{x}^{(0)}) = 0$$

y como el error inicial es arbitrario, del lema 2 tenemos que $\rho(\mathbf{B}) < 1$

Métodos Iterativos

Convergencia

Por el lema 3 tenemos que

Corolario: Una condición suficiente para que un método iterativo estacionario $\mathbf{x}^{(k)} = \mathbf{B}\mathbf{x}^{(k-1)} + \mathbf{c}$ converja para toda aproximación inicial $\mathbf{x}^{(0)}$ es que $\| \mathbf{B} \| < 1$ para alguna norma matricial inducida; y el error está acotado por

- $\| \mathbf{x} - \mathbf{x}^{(k)} \| \leq \| \mathbf{B} \|^k \| \mathbf{x} - \mathbf{x}^{(0)} \|$
- $\| \mathbf{x} - \mathbf{x}^{(k)} \| \leq \frac{\|\mathbf{B}\|}{1-\|\mathbf{B}\|} \| \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \|$

La segunda cota viene de

$$\mathbf{x} - \mathbf{x}^{(k)} = \mathbf{B}(\mathbf{x} - \mathbf{x}^{(k-1)}) = \mathbf{B}(\mathbf{x} - \mathbf{x}^{(k)}) - \mathbf{B}(\mathbf{x}^{(k)} + \mathbf{x}^{(k-1)})$$

$$\| \mathbf{x} - \mathbf{x}^{(k)} \| \leq \| \mathbf{B} \| \| \mathbf{x} - \mathbf{x}^{(k)} \| + \| \mathbf{B} \| \| \mathbf{x}^{(k)} + \mathbf{x}^{(k-1)} \|$$

Métodos Iterativos

Convergencia

La convergencia de los métodos iterativos estacionarios es lineal

$$\| \mathbf{x} - \mathbf{x}^{(k)} \| \leq \| \mathbf{B} \| \| \mathbf{x} + \mathbf{x}^{(k-1)} \|$$

El criterio de parada viene dado por

$$\frac{\| \mathbf{B} \|}{1 - \| \mathbf{B} \|} \| \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \| \leq \epsilon$$

y para estimar el valor de $\| \mathbf{B} \|$ tenemos que para k grande

$$\| \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \| \approx \| \mathbf{B} \| \| \mathbf{x}^{(k-1)} - \mathbf{x}^{(k-2)} \|$$

por lo que

$$\| \mathbf{B} \| \approx \frac{\| \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \|}{\| \mathbf{x}^{(k-1)} - \mathbf{x}^{(k-2)} \|}$$

Metodo de Relajación

En términos del vector residual el método de Gauss-Seidel se puede escribir como

$$x_i^{(k+1)} = x_i^{(k)} + r_i^{(k)} \quad , \quad i = 1, 2, \ldots, n$$

con

$$r_i^{(k)} = -\left(\sum_{j=1}^{i-1} a_{ij} x_j^{(k)} + \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \right) / a_{ii} \quad , \quad \begin{array}{l} i = 1, 2, \ldots, n \\ k = 0, 1, \ldots \end{array}$$

Si modificamos el método anterior a

$$x_i^{(k+1)} = x_i^{(k)} + \omega r_i^{(k)} \quad , \quad i = 1, 2, \ldots, n$$

cuya forma matricial es

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \omega(-\mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{U}\mathbf{x}^{(k)} - \mathbf{I}\mathbf{x}^{(k)}) + \omega \mathbf{D}^{-1}\mathbf{b} \quad ,$$

obtenemos

Metodo de Relajación

$$\mathbf{B}_\omega = (\mathbf{I} + \omega\mathbf{L})^{-1}\left((1-\omega)\mathbf{I} - \omega\mathbf{U}\right) \quad , \quad \mathbf{c}_\omega = (\mathbf{I} + \omega\mathbf{L})^{-1}\omega\mathbf{D}^{-1}\mathbf{b}$$

Para ciertos valores de ω la convergencia del método mejora

- Para $0 < \omega < 1$, el método se llama de subrelajación y logra converger para algunos casos en los que el método de Gauss-Seidel diverge
- Para $\omega > 1$, el metodo se llama de sobrerelajación o S.O.R *Succesive Over-Relaxation* y acelera la convergencia

Teorema de Kahan: Si $a_{ii} \neq 0 \; \forall \; i$, entonces $\rho(\mathbf{B}_\omega) \geq |\omega - 1|$. Es decir que para que el método converja, $\rho(\mathbf{B}_\omega) < 1$, nesesariamente $0 < \omega < 2$.

Demostración: Por un lado tenemos que

$$\det(\mathbf{B}_\omega) = \lambda_1\lambda_2\cdots\lambda_n \quad ,$$

donde, $\lambda_1, \lambda_2, \ldots, \lambda_n$; son los autovalores de $\mathbf{B}_\omega$

Metodo de Relajación

Por otro lado,

$$\begin{aligned} \det(\mathbf{B}_\omega) &= \det\left((\mathbf{I}+\omega\mathbf{L})^{-1}\left((1-\omega)\mathbf{I}-\omega\mathbf{U}\right)\right) \\ &= \det\left((\mathbf{I}+\omega\mathbf{L})^{-1}\right)\det\left((1-\omega)\mathbf{I}-\omega\mathbf{U}\right) \\ &= (1-\omega)^n \end{aligned}$$

de lo anterior tenemos que

$$|1-\omega|^n = |\lambda_1||\lambda_2|\cdots|\lambda_n| \leq \rho(\mathbf{B}_\omega)^n \leq 1 \quad ,$$

es decir,

$$0 < \omega < 2$$

Teorema de Ostrowski-Reich: Si **A** es una matriz $n \times n$ definida positiva y $0 < \omega < 2$, entonces el método de relajación converge para cualquier $\mathbf{x}^{(0)} \in \mathbb{R}^n$

Metodo de Relajación

Teorema: Si **A** es una matriz definida positiva y tridiagonal, entonces $\rho(\mathbf{B}_{GS}) = (\rho(\mathbf{B}_J))^2 < 1$, y el valor óptimo de ω para el método de relajación es

$$\omega = \frac{2}{1 + \sqrt{1 - (\rho(\mathbf{B}_J))^2}}$$

Con lo que

$$\rho(\mathbf{B}_\omega) = \omega - 1$$

Método de Relajación

```
Entrada: n,A[n,n],b[n],w,x[n],t,m
Salida: x[n], e
sor(n, A, b, w, t, m, x, e)
k ← 0
repita
| k ← k + 1
| para i ← 1 hasta n
| | xa[i] ← x[i]
| para i ← 1 hasta n
| | x[i] ← b[i]
| | para j ← 1 hasta i-1
| | | x[i] ← x[i]
| | |      - A[i,j]*x[j]
| | para j ← i+1 hasta n
| | | x[i] ← x[i]
| | |      - A[i,j]*xa[j]
| | x[i] ← (1-w)*xa[i]
| |      + w*x[i]/A[i,i]
| para i ← 1 hasta n
| | d[i] ← x[i] - xa[i]
| si(norma(n,d)> tol)
| | e ← verdadero
| sino
| | e ← falso
hasta((¬ e) ∨ (k>m))
retornar
```

Note que para la condición de parada estamos usando la función norma que calcula la norma vectorial de d

Errores

Se puede demostrar[a] que si usamos aleminación gaussiana con aritmética de t dígitos, que el vector residual $\mathbf{r}$ de la solución aproximada $\tilde{\mathbf{x}}$ de $\mathbf{Ax} = \mathbf{b}$ puede aproximarse a

$$\mathbf{r} \approx 10^{-t} \parallel \mathbf{A} \parallel \parallel \tilde{\mathbf{x}} \parallel$$

Para calcular una aproximación del número de condición $\mathcal{K}(\mathbf{A})$ resolvemos el sistema $\mathbf{Ay} = \mathbf{r}$, cuya solución aproximada

$$\tilde{\mathbf{y}} \approx \mathbf{A}^{-1}\mathbf{r} = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}) = \mathbf{A}^{-1}\mathbf{b} - \mathbf{A}^{-1}\mathbf{A}\tilde{\mathbf{x}} = \mathbf{x} - \tilde{\mathbf{x}} \quad ;$$

lo que implica que

$$\parallel \tilde{\mathbf{y}} \parallel \approx \parallel \mathbf{x} - \tilde{\mathbf{x}} \parallel \leq \parallel \mathbf{A}^{-1} \parallel \parallel \mathbf{r} \parallel \approx 10^{-t} \underbrace{\parallel \mathbf{A}^{-1} \parallel \parallel \mathbf{A} \parallel}_{\mathcal{K}(\mathbf{A})} \parallel \tilde{\mathbf{x}} \parallel$$

[a] Forsythe G.E., Moler C.B., *Computer solution of linear algebraic systems*, Prentice-Hall, Englewood Cliffs, NJ, 1967, 148 pp

Errores

de donde obtenemos que

$$\mathcal{K}(\mathbf{A}) \approx 10^{-t} \frac{\| \tilde{\mathbf{y}} \|}{\| \tilde{\mathbf{x}} \|}$$

Ejemplo:
En el sistema lineal siguiente tiene por solución a $\mathbf{x} = (1,1)^T$

$$\begin{pmatrix} 1 & 2999 \\ 1 & 6666 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3000 \\ 6667 \end{pmatrix} \quad \Rightarrow$$

Aplicando eliminación de Gauss con aritmética de $t = 5$ dígitos

$$\begin{pmatrix} 0.5501 & 0 \\ 1 & 6666 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.49489 \\ 6667 \end{pmatrix}$$

Errores

La solución aproximada del sistema es $\tilde{\mathbf{x}} = (0.89964, 1)^T$
El vector resudual correspondiente a $\tilde{\mathbf{x}}$ es

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}} = \begin{pmatrix} -0.10036 \\ -0.10036 \end{pmatrix} \quad , \quad \| \mathbf{r} \|_\infty = 0.10036$$

Para estimar el número de condición resolvemos el sistema

$$\mathbf{A}\tilde{\mathbf{y}} = \mathbf{r} \quad \text{, obteniendo} \quad \tilde{\mathbf{y}} = \begin{pmatrix} 0.10036 \\ 0 \end{pmatrix}$$

Utilizando la estimación tenemos que

$$\mathcal{K}(\mathbf{A}) \approx 10^5 \frac{\| \tilde{\mathbf{y}} \|_\infty}{\| \tilde{\mathbf{x}} \|_\infty} = 10036 \quad ; \quad \mathcal{K}(\mathbf{A}) = 12117.69$$

Método de Refinamiento Iterativo

Para el cálculo aproximado del número de condición estimamos el error mediante la expresión

$$\mathbf{x} - \tilde{\mathbf{x}} \approx \tilde{\mathbf{y}}$$

El método consiste en calcular a partir de un valor inicial de $\tilde{\mathbf{x}}$ el vector residual $\mathbf{r}$, luego con este último calculamos el vector $\tilde{\mathbf{y}}$ y finalmente usamos a $\tilde{\mathbf{y}}$ para calcular una nueva solución aproximada $\tilde{\mathbf{x}}$

$$\begin{aligned} \mathbf{r}^{(k)} &= \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} \\ \mathbf{LU}\mathbf{y}^{(k)} &= \mathbf{r}^{(k)} \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \mathbf{y}^{(k)} \end{aligned}$$

En general, $\tilde{\mathbf{x}} + \tilde{\mathbf{y}}$ aproxima mejor a $\mathbf{x}$ que $\tilde{\mathbf{x}}$.

Método de Refinamiento Iterativo

Comentarios:

- Una vez factorizada la matriz $\mathbf{A} = \mathbf{LU}$ en cada iteración se requieren $2n^2$ operaciones
- Si el sistema está bien condicionado con una o dos iteraciones se obtiene la solución exacta.
- Con aritmética de t dígitos, un sistema con $\mathcal{K}(\mathbf{A}) \approx 10^q$ luego de k iteraciones el número de dígitos correctos en la solucion es aproximadamente

$$\min(t, k(t-q))$$

- En general la solución en sistemas mal condicionados puede mejorarse significativamente excepto cuando $\mathcal{K}(\mathbf{A}) > 10^t$

Teoría de Interpolación

- Interpolación polinomial
- Forma de Lagrange del polinomio interpolatorio
- Diferencias divididas.
  Forma de Newton del polinomio de interpolación
- Diferencias progresivas y regresivas.
- Diferencias divididas con nodos repetidos
- Convergencia de los polinomios interpolatorios.
- Nodos de Polinomios de Chebyshev
- Interpolación con funciones splines.
  Splines cúbicas

Interpolación

Definición: proceso de obtención de nuevos puntos partiendo del conocimiento de un conjunto discreto de puntos.
Problemas:

- Teniendo de cierto número de puntos, obtenidos por muestreo o a partir de un experimento, se quiere construir una función que los ajuste.
- Aproximación de una función complicada, es decir una función cuyo cálculo resulta costoso; por una más simple. Esto se logra partiendo de un cierto número de sus valores e interpolar dichos datos construyendo una función más simple

En general, se trata de obtener una función $f(x)$, denominade **función interpolante**, a partir de n puntos distintos (x_k, y_k), llamados **nodos**, tal que

$$f(x_k) = y_k \quad , \quad k = 1, \ldots, n$$

Interpolación de Taylor

Usa el desarrollo de Taylor de una función en un punto, x_0 para construir un polinomio de grado n que la aproxima.

$$P(x) = \sum_{i=1}^{n} \frac{f^{(i)}(x_0)}{i!}(x - x_0)^i \quad ; \quad \varepsilon(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)^{n+1}$$

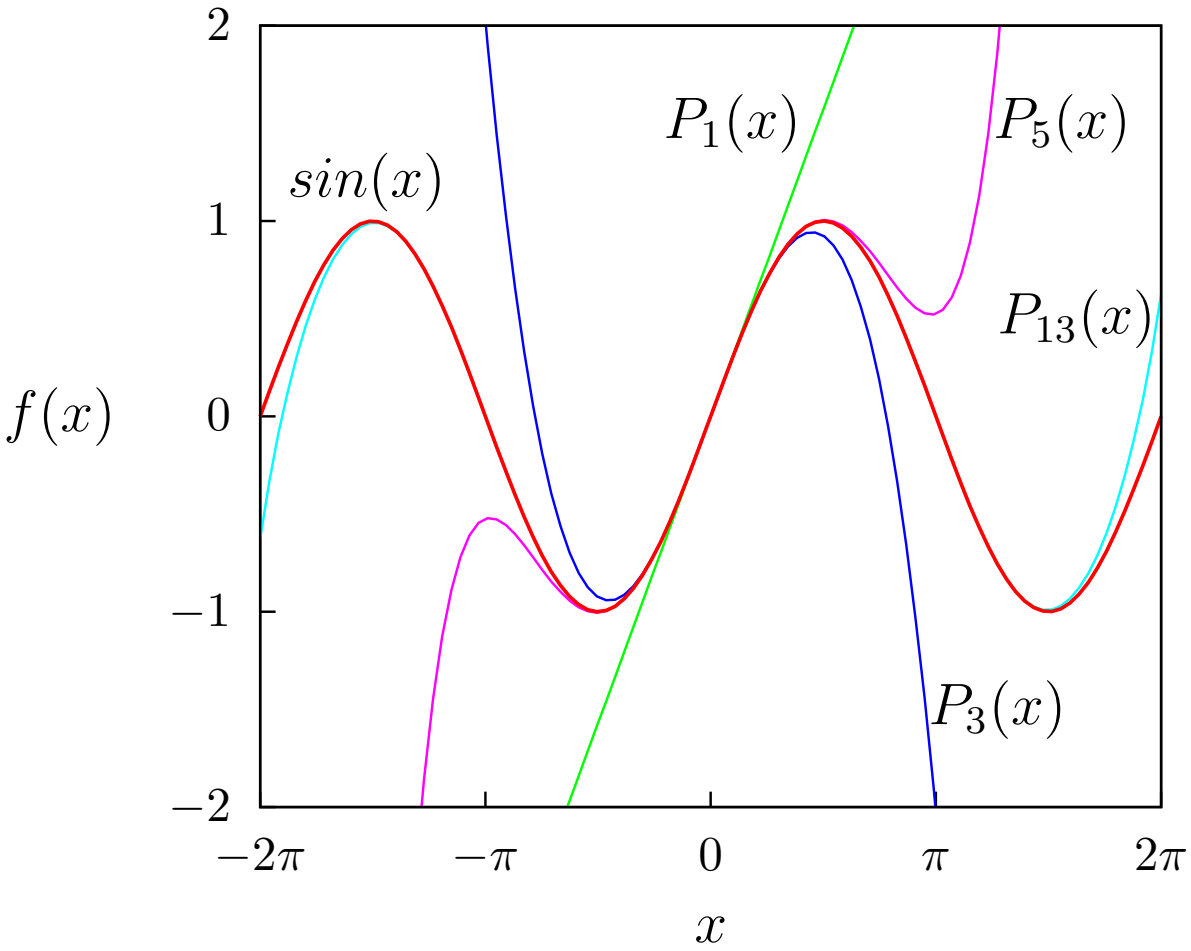
Ventajas:

- Sólo necesita de un punto x_0 de la función
- El cálculo es sencillo comparado con otras formas de interpolación polinómica

Desventajas:

- Se debe conocer a la función
- Requiere que la función sea suficientemente diferenciable en un entorno a x_0
- El error de interpolación por lo general crece al alejarse de x_0

Interpolación de Taylor



$$f(x) = \sin(x)$$

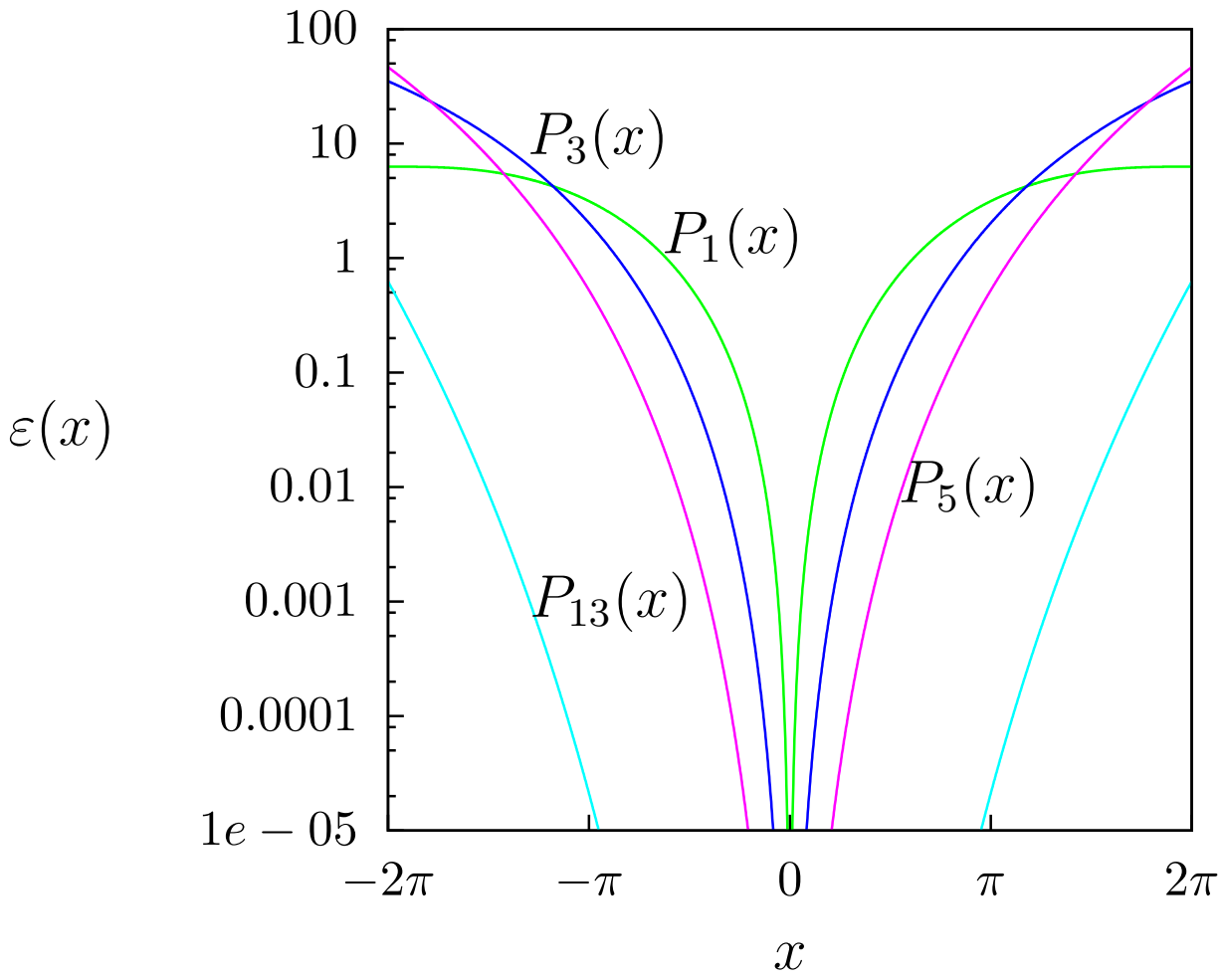
$$P_1(x) = x$$

$$P_3(x) = x - \frac{x^3}{6}$$

$$P_5(x) = x - \frac{x^3}{6} + \frac{x^5}{120}$$

Interpolación de Taylor

$$\varepsilon(x) = |\sin(x) - P_n(x)|$$



Interpolación Lineal

Es la más simple de las interpolaciones.

Supongamos que se quiere calcular el valor una función $f(\bar{x})$, de la cual conocemos un conjunto de puntos $\{(x_i, f(x_i)\}, i = 0, 1, \ldots, m$; con

$$x_i < x_{i+1} \; \forall \; i < m, \quad \text{y} \quad x_i < \bar{x} < x_{i+1}$$

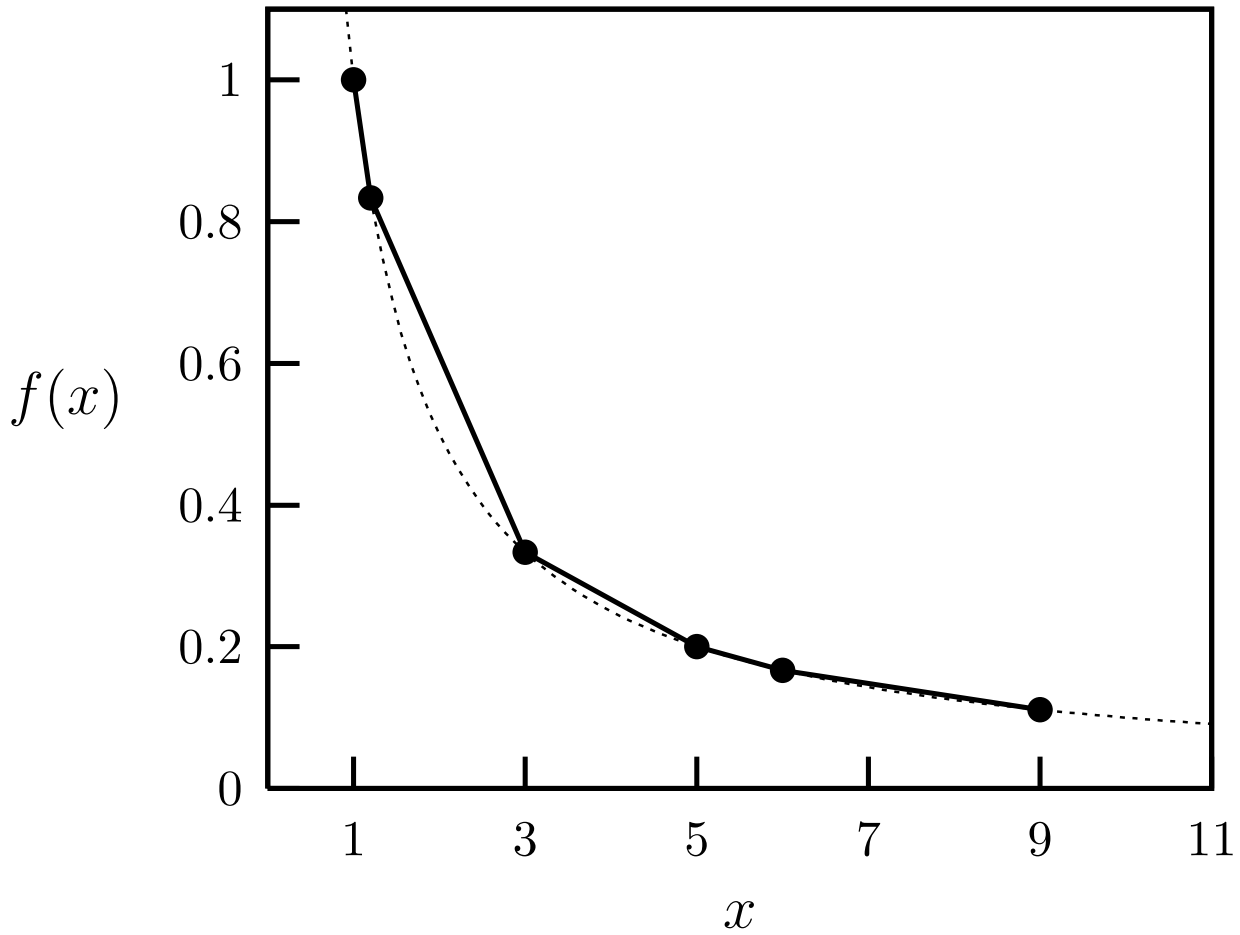
El polinomio interpolatorio lineal que aproxima a $f(\bar{x})$ viene dado por la expresión

$$P(x) = f(x_i) + \frac{x - x_i}{x_{i+1} - x_i} \left(f(x_{i+1}) - f(x_i)\right) \quad \text{, reordenando}$$

$$\begin{aligned} P(x) &= \overbrace{\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}}^{a_1} x + \overbrace{f(x_i) - \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}}^{a_0} \\ P(x) &= a_1 x + a_0 \end{aligned}$$

Interpolación Lineal

$$f(x) = \frac{1}{x}$$



Interpolación Polinomial

En general, una mejor aproximación a $f(x)$ se puede obtener con un polinomio de grado mayor.

Teorema: Dados $n+1$ puntos distintos $\{(x_i, f(x_i)\}, i = 0, 1, \ldots, n$ existe un único polinomio de grado $\leq n$ que interpola a $f(x)$

Demostración: Supongamos que existen dos polinomios $P(x)$ y $Q(x)$ que interpolan al conjunto de puntos. la diferencia entre los dos polinomios viene dada por

$$R(x) = P(x) - Q(x) \ ,$$

donde, $R(x)$ es un polinomio de grado $\leq n$. Como $f(x_i) = P(x_i) = Q(x_i),\ i = 0, 1, \ldots, n$; entonces $R(x)$ tiene $n+1$ ceros, sin embargo como el grado de $R(x)$ es a lo sumo n solamente podría tener n ceros, Así que $R(x)$ no puede existir a menos que sea igual a cero. Es decir $P(x) = Q(x)$

Interpolación Polinómica

Conocidos $m+1$ puntos, $x_0, x_1, \ldots x_m$; si se usan:

- 2 puntos $\Rightarrow P(x) = a_0 + a_1 x$
- 3 puntos $\Rightarrow P(x) = a_0 + a_1 x + a_2 x^2$
- 4 puntos $\Rightarrow P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$
- ...
- $n+1$ puntos con $n \leq m \Rightarrow$

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + a_n x^n$$

A diferencia del caso lineal, el cálculo de los $n+1$ coeficientes de los polinomios de grado $n > 1$ no es tan fácil de calcular

Existen diferentes métodos para el cálculo de los coeficientes

Método Directo

Con los $n+1$ puntos podemos plantear el sistema de ecuaciones

$$\begin{array}{rcccl} P_n(x_0) & = & a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_{n-1} x_0^{n-1} + a_n x_0^n & = & f(x_0) \\ P_n(x_1) & = & a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_{n-1} x_1^{n-1} + a_n x_1^n & = & f(x_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P_n(x_n) & = & a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_{n-1} x_n^{n-1} + a_n x_n^n & = & f(x_n) \end{array}$$

Note que las $n+1$ incognitas son los coeficientes de los polinomios a_i y no las x_i, como es lo acostumbrado La matriz del sistema, $\mathbf{V}_m$[a], viene dada por

$$\mathbf{V}_n \mathbf{a} = \mathbf{f}(\mathbf{x}) \quad \text{donde} \quad \mathbf{V}_n = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix}$$

[a] Llamada matriz de Vandermonde

Método de Lagrange

El polinomio interpolatorio de Lagrange tiene la forma

$$P_n(x) = f(x_0)L_0(x)+f(x_1)L_1(x)+\cdots+f(x_n)L_n(x) = \sum_{i=0}^{n} f(x_i)L_i(x) \quad ,$$

donde, $L_i(x),\ i = 0, 1, \ldots, n$ son polinomios construidos para que

$$L_i(x_j) = \begin{cases} 1 & , \text{ si } i = j \\ \\ 0 & , \text{ si } j \neq i \end{cases}$$

Una forma de construirlos es usando la expresión

$$L_i(x) = \frac{(x - x_0)(x - x_1)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n)}{(x_i - x_0)(x_i - x_1)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)}$$

Los $L_i(x)$ se llaman polinomios de Lagrange

Método de Lagrange

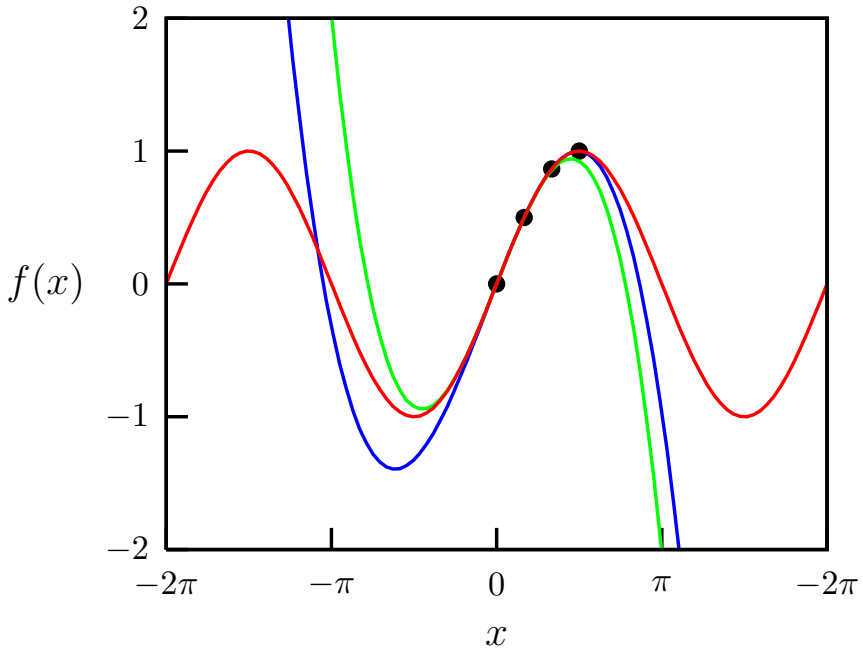
Ejemplo: Conocidos los puntos $(0,0); (\pi/6, 1/2); (\pi/3, \sqrt{3}/2)$ y $(\pi, 1)$; calcular el polinomio interpolador de grado 3 que pasa por los 4 punto.

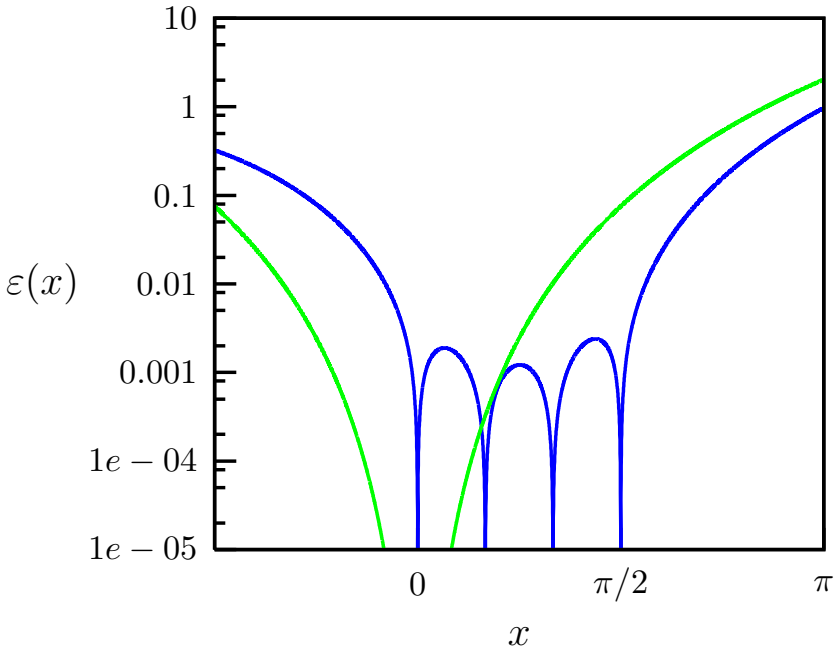
$$P(x) = \frac{(x-\pi/6)(x-\pi/3)(x-\pi)}{(0-\pi/6)(0-\pi/3)(0-\pi)}0 + \frac{(x-0)(x-\pi/3)(x-\pi)}{(\pi/6-0)(\pi/6-\pi/3)(\pi/6-\pi)}\frac{1}{2}$$

$$+ \frac{(x-0)(x-\pi/6)(x-\pi)}{(\pi/3-0)(\pi/3-\pi/6)(\pi/3-\pi)}\frac{\sqrt{3}}{2} + \frac{(x-0)(x-\pi/6)(x-\pi/3)}{(\pi-0)(\pi-\pi/6)(\pi-\pi/3)}1$$

$$P(x) = 1.0204287x - 0.065470801x^2 - 0.11387190x^3$$

Método de Lagrange

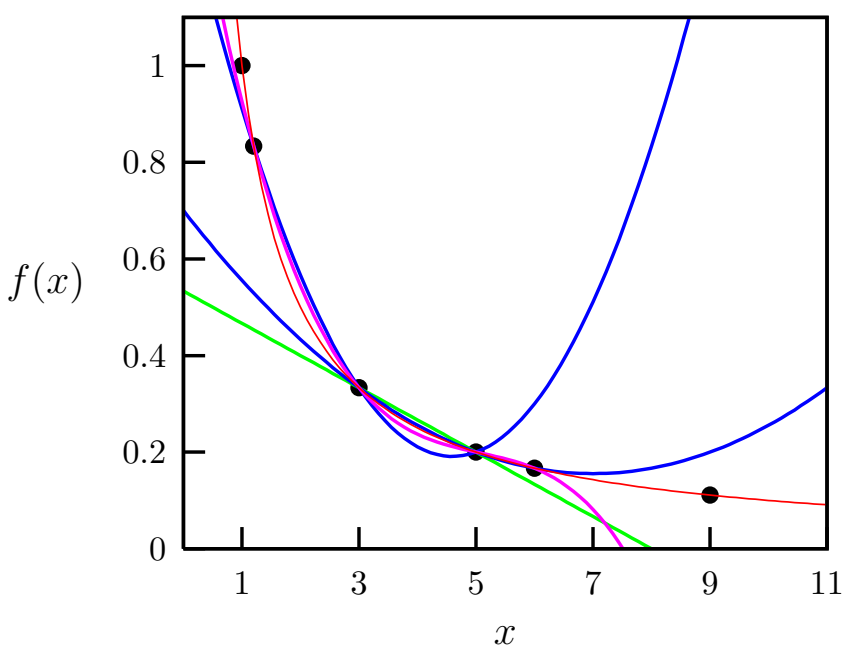


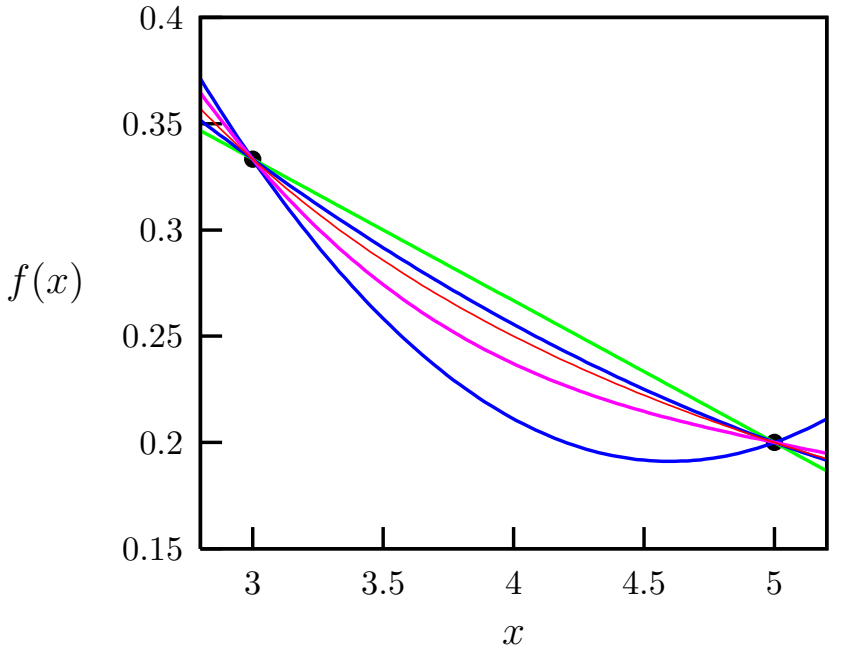


- Rojo $f(x) = sin(x)$.
- Azul polinomio interpolatorio de grado 3.
- Verde Polinomio de Taylor de grado tres en $x_0 = 0$

Método de Lagrange

Ejemplo: dados los puntos $(1, 1)$, $(1.2, 0.8333)$, $(3, 0.3333)$, $(5, 0.2)$, $(6, 0.1667)$ y $(9, 0.1111)$ calcular el valor de la función en $\bar{x} = 4$





- Rojo $f(x) = 1/x$.
- Verde $P_1(x)$.
- Azul $P_2(x)$.
- Violeta $P_3(x)$

Error del polinomio interpolatorio

Teorema: Sean $f(x)$ una función $(n+1)$ veces continuamente diferenciable en el intervalo $I = (\min_{i=0:n}(x_i), \max_{i=0:n}(x_i))$; $P_n(x)$ el polinomio de interpolación de $f(x)$ en los puntos $x_0, x_1, \ldots, x_n$; y $\bar{x}$ un punto en el intervalo I. Entonces existe un $\xi \in I$ tal que

$$\varepsilon(\bar{x}) = f(\bar{x}) - P_n(\bar{x}) = f^{n+1}(\xi)\frac{\phi(\bar{x})}{(n+1)!}$$

donde, $\phi(x) = (x - x_0)(x - x_1)\cdots(x - x_n)$.
Demostración: Si $\bar{x} = x_i$ el error $\varepsilon(\bar{x})$ y no es necesario interpolar. Para $\bar{x} \neq x_i$, $i = 0, 1, \ldots, n$ definamos una nueva función $g(x)$ tal que

$$g(x) = f(x) - P_n(x) - \phi(x)\frac{f(\bar{x}) - P_n(\bar{x})}{\phi(\bar{x})}$$

La función $g(x)$ tiene $(n+2)$ raices en el intervalo I:

$$g(x_i) = 0\ ,\ i = 0, 1, \ldots, n \quad \text{y} \quad g(\bar{x}) = 0$$

Error del polinomio interpolatorio

Note que $\phi(\bar{x}) \neq 0$ y $f(x), P_n(x)$ y $\phi(x)$ son continuas y diferenciables entonces $g(x)$ es contínua y diferenciable podemos aplicar el

Teorema de Rolle o del Valor Medio: Sea $g(x)$ y su derivada $g'(x)$ continuas en el intervalo $[a, b]$. Entonces existe al menos un valor $x = \xi$ en $[a, b]$ tal que

$$g'(\xi) = \frac{g(b) - g(a)}{b - a} \ .$$

Así que si $g(x)$ tiene $(n + 2)$ ceros en I, entonces $g'(x)$ tendrá $(n + 1)$ ceros en I, $g''(x)$ tendrá (n) ceros en I y así sucesivamente hasta $g^{(n+1)}(x)$ que tendrá 1 ceros en I. Llamemos ξ a este punto donde

$$g^{(n+1)}(\xi) = 0$$

Error del polinomio interpolatorio

Derivando $(n+1)$ veces a $g(x)$ respecto a x tenemos

$$g^{(n+1)}(x) = f^{(n+1)}(x) + \underbrace{P_n^{(n+1)}(x)}_{0} + \underbrace{\phi^{(n+1)}(x)}_{(n+1)!} \frac{f(\bar{x}) - P_n(\bar{x})}{\phi(\bar{x})}$$

Evaluando en $x = \xi$ tenemos

$$g^{(n+1)}(\xi) = f^{(n+1)}(\xi) + (n+1)! \frac{f(\bar{x}) - P_n(\bar{x})}{\phi(\bar{x})} = 0 \ ,$$

de donde obtenemos

$$\varepsilon(\bar{x}) = f(\bar{x}) - P_n(\bar{x}) = f^{(n+1)}(\xi) \frac{\phi(\bar{x})}{(n+1)!}$$

Solamente se puede usar en funciones donde la $(n+1)$ derivada tiene una cota conocida en el intervalo I.

Método de Newton

Algorítmicamente, el método de Lagrange, presenta un problema: para pasar de un polinomio de interpolación de grado n a uno de grado $(n+1)$ hay que iniciar los cálculos desde el principio, es decir que no se pueden utilizar los coeficientes de P_n para calcular los de P_{n+1}.

El método de Newton, que es algorítmicamente más eficiente, construye el polinomio de interpolación P_n cuando se conoce el polinomio de interpolación de grado $(n-1)$.

$$\begin{aligned} P_n(x) &= \overbrace{a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots}^{P_{n-1}(x)} + \\ &\quad a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1}) \end{aligned}$$

$$P_n(x) = P_{n-1}(x) + a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})$$

Método de Newton

Usando la notación de diferencias divididas[a] tenemos

$$
\begin{aligned}
P_0(x) &= f(x_0) = f[x_0] \\
P_1(x) &= f(x_0) + \frac{f(x_1)-f(x_0)}{x_1-x_0}(x-x_0) = f[x_0] + f[x_0,x_1](x-x_0) \\
P_2(x) &= f(x_0) + \frac{f(x_1)-f(x_0)}{x_1-x_0}(x-x_0) + \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2-x_0}(x-x_0)(x-x_1) \\
&= f[x_0] + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1)
\end{aligned}
$$

donde,

$$a_0 = f[x_0] \quad , \quad a_1 = f[x_0,x_1] \quad , \quad a_2 = f[x_0,x_1,x_2]$$

[a] Estudiada en el método de bisección para el cálculo de raices

Método de Newton

Siguiendo con polinomios de grado cada vez mayor obtenemos

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

donde,

$$a_n = f[x_0, x_1, \ldots, x_n]$$

Método de Newton

Ejemplo: dados los puntos $(1,1)$, $(1.2, 0.8333)$, $(3, 0.3333)$, $(5, 0.2)$, $(6, 0.1667)$ y $(9, 0.1111)$ calcular el valor de la función en $\bar{x} = 4$

x	f	$f[,]$	$f[,,]$	$f[,,,]$	$f[,,,,]$	$f[,,,,,]$
1	1					
		-4.167				
1.2	0.8333		0.6945			
		-2.778		-0.09425		
3	0.3333		0.3175		0.006084	
		-0.06665		-0.06383		0.0002426
5	0.2		0.01112		0.008025	
		-0.0333		-0.001238		
6	0.1667		0.003693			
		-0.01853				
9	0.1111					

Método de Newton

Diferencias Divididas

Entrada: `f[n], x[n], n`
Salida: `a[n]`

```
difDiv(f, x, n, a)
para i ← 0 hasta n
| a[i] ← f[i]
para i ← 1 hasta n
| para j ← n hasta i
| | a[j] ← (a[j]-a[j-1])
| |       /(x[j]-x[j-i])
```

Polinomio de Newton

Entrada: `a[n], x[n], n, xx`
Salida: `Pn`

```
poliInter(a, x, n, xx,Pn)
Pn ← a[n]
para i ← n-1 hasta 0
| Pn ← a[i] + Pn*(xx - x[i])
```

El subprograma `poliInter()` recibe como entrada al vector `a[]` que previamente calcula el subprograma `difDiv`

Método de Newton

Cuando los nodos están igualmente espaciados podemos usar el operador de

Diferencias Progresivas	Diferencias Regresivas
Definida como	Definida como
$\Delta f(x) = f(x+h) - f(x)$	$\nabla f(x) = f(x) - f(x-h)$
de orden $k \geq 1$ se define	de orden $k \geq 1$ se define
$\Delta^k f(x) = \Delta^{k-1} f(x+h) - \Delta^{k-1} f(x)$	$\nabla^k f(x) = \nabla^{k-1} f(x) - \nabla^{k-1} f(x-h)$
con	con
$\Delta^0 f(x) = f(x)$	$\nabla^0 f(x) = f(x)$
por lo que tenemos	por lo que tenemos
$\Delta^k f(x) = \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} f(x+jh)$	$\nabla^k f(x) = \sum_{j=0}^{k} (-1)^{j} \binom{k}{j} f(x-jh)$
Lema:	**Lema:**
$f[x_0, x_1, \ldots, x_k] = \frac{\Delta^k f(x_0)}{k! h^k}$	$f[x_{n-k}, \ldots, x_{n-1}, x_n] = \frac{\nabla^k f(x_n)}{k! h^k}$

donde, $k \geq 0$

Método de Newton

Diferencias Progresivas

Prueba del Lema: Para $k = 0$ tenemos

$$f[x_0] = f(x_0) = \frac{\Delta^0 f(x_0)}{0!h^0}$$

Para $k = 1$ tenemos

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f(x+h) - f(x)}{h} = \frac{\Delta^1 f(x_0)}{1!h^1}$$

Supongamos que cierta la k y probemos para $k+1$

$$f[x_0, x_1, \ldots, x_{k+1}] = \frac{f[x_1, x_2, \ldots, x_{k+1}] - f[x_0, x_1, \ldots, x_k]}{x_{k+1} - x_0} =$$

$$\left(\frac{\Delta^k f(x_1)}{k!h^k} - \frac{\Delta^k f(x_0)}{k!h^k} \right) /(r+1)h = \frac{\Delta^k f(x_0 + h) - \Delta^k f(x_0)}{(r+1)!h^{k+1}} = \frac{\Delta^{k+1} f(x_0)}{(r+1)!h^{k+1}}$$

Método de Newton

Diferencias Progresivas: Sea $x = x_0 + sh$ entonces $x - x_i = (s-i)h$

$$\begin{aligned} P_n(x) &= f[x_0] + f[x_0,x_1]\underbrace{s}_{\binom{s}{1}1!}h + f[x_0,x_1,x_2]\underbrace{s(s-1)}_{\binom{s}{2}2!}h^2 + \\ &\quad f[x_0,x_1,x_2,x_3]\underbrace{s(s-1)(s-2)}_{\binom{s}{3}3!}h^3 + \cdots + \\ &\quad f[x_0,\ldots,x_k]\underbrace{s(s-1)\ldots(s-k+1)}_{\binom{s}{k}k!}h^k + \cdots + \\ &\quad f[x_0,\ldots,x_n]\underbrace{s(s-1)\cdots(s-n+1)}_{\binom{s}{n}n!}h^n \end{aligned}$$

Entonces,

$$P_n(x) = P_n(x_0 + sh) = \sum_{k=0}^{n} \binom{s}{k} \Delta^k f(x_0)$$

Método de Newton

Diferencias Regresivas: Como tarea demuestre en primer lugar el lema para el cálculo de las diferencias regresivas y luego demuestre que el polinomio de interpolación de Newton para diferencias regresivas tiene la forma

$$P_n(x) = P_n(x_n - sh) = \sum_{k=0}^{n} (-1)^k \binom{-s}{k} \nabla^k f(x_k)$$

Fenómeno de Runge

Runge descubrió que el polinomio de interpolación $P_n(x)$ de algunas funciones construido usando puntos equidistantes oscila hacia los extremos del intervalo que si se interpola. Este fenómeno lo observó con la función

$$f(x) = \frac{1}{1+25x^2}$$

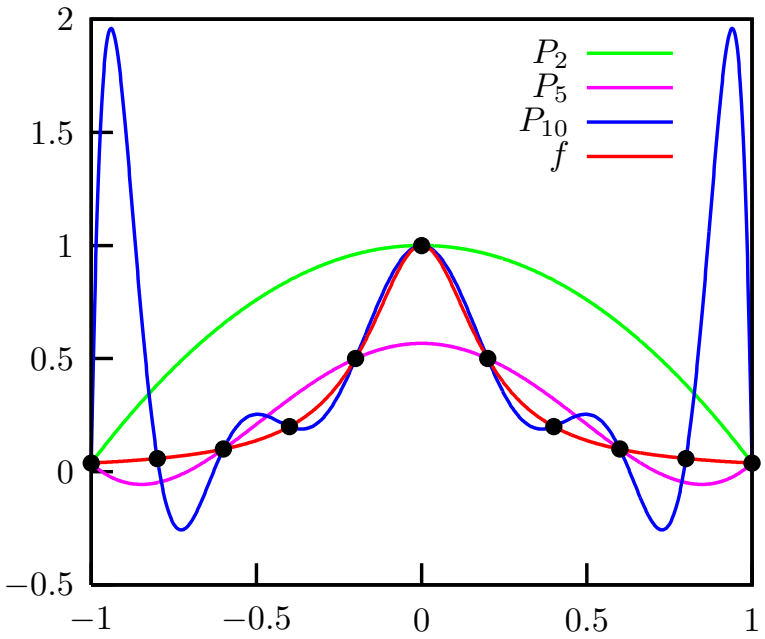
usando puntos equidistantes $x_i \in [-1,1]$ tal que

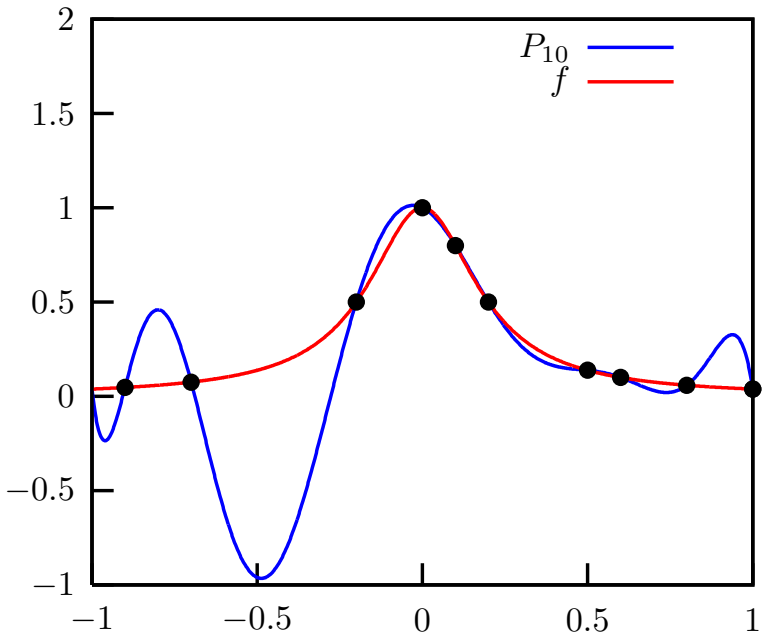
$$x_i = -1 + i\frac{2}{n}, \qquad i \in \{0,1,2,\ldots,n\}$$

Con esta función, el error de interpolación tiende a infinito cuando crece el grado del polinomio:

$$\lim_{n\to\infty}\left(\max_{-1\leq x\leq 1}|f(x)-P_n(x)|\right)=\infty.$$

Fenómeno de Runge





El error entre la función $f(x)$ y el polinomio de interpolación $P_n(x)$ está limitado por el n-esima derivada de $f(x)$. Para el caso de la función de Runge la magnitud del valor máximo de las derivadas en el intervalo $[-1, 1]$ aumenta al incrementarse el orden, por lo que la cota del error crece a medida que aumenta el grado de $P_n(x)$

Fenómeno de Runge

- La oscilación se puede minimizar usando nodos de Chebyshev en lugar de nodos equidistantes. En este caso se garantiza que el error máximo disminuye al crecer el orden polinómico.
- El fenómeno demuestra que los polinomios de grado alto no son, en general, aptos para la interpolación.
- El problema se puede evitar usando curvas spline que son polinomios por partes. Cuando se intenta reducir el error de interpolación se puede incrementar el número de partes del polinomio que se usan para construir la s pline, en lugar de incrementar su grado.

Nodos de Chebyshev

El polinomio de interpolación que resulta del uso de los nodos de Chebyshev reduce al mínimo el problema generado por el fenómeno de Runge.
Los nodos de Chebyshev

$$x_k = cos\left(\frac{\pi(2k+1)}{2(n+1)}\right) \quad , \quad k = 0, 1, \ldots, n$$

son las raíces del polinomio de Chebyshev de primer tipo

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad , \quad n > 2 \quad \text{con} \quad T_0(x) = 1 \quad \text{y} \quad T_1(x) = x$$

Teorema:

$$T_n(x) = \cos(n \arccos(x)) \quad , \quad n \geq 0 \text{ y } x \in [-1, 1]$$

de donde

$$T_n(\cos(\theta)) = \cos(n\theta) \quad , \quad n = 0, 1, \ldots$$

Nodos de Chebyshev

Demostración: Como

$$\cos((n+1)\theta) = \cos(\theta)\cos(n\theta) - sin(\theta)\sin(n\theta) \quad \text{y}$$
$$\cos((n-1)\theta) = \cos(\theta)\cos(n\theta) + sin(\theta)\sin(n\theta)$$

tenemos que

$$\cos((n+1)\theta) = 2\cos(\theta)\cos(n\theta) - \cos((n-1)\theta)$$

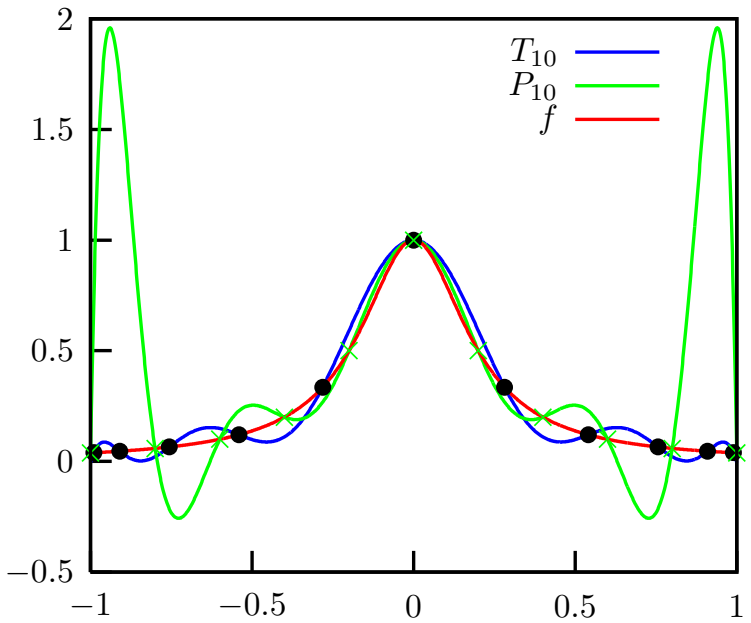
Definamos $f_n(x) = \cos(n \arccos(x))$ con $x = cos(\theta)$ entonces

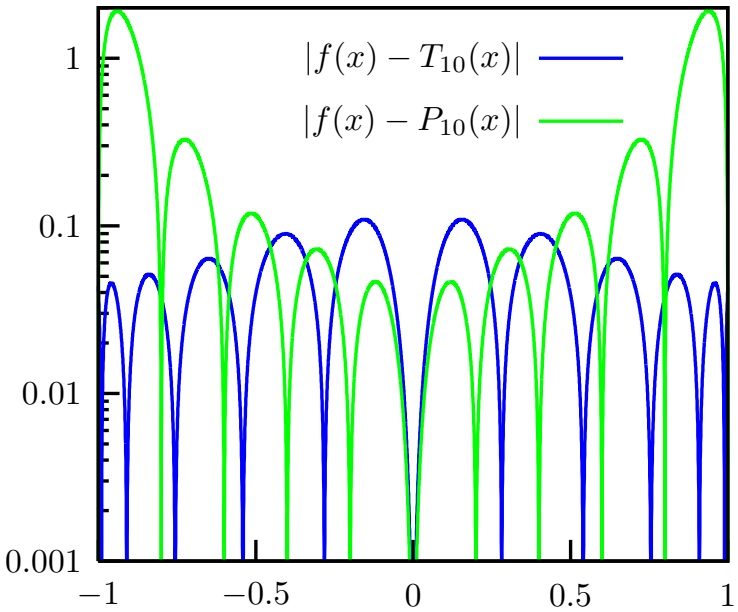
$$\begin{aligned} f_0(x) &= cos(0 \times \theta) = 1 \quad , \quad f_1(x) = cos(\theta) = x \quad , \\ f_{n+1} &= \cos((n+1)\arccos(\cos(\theta))) = \cos((n+1)\theta) \\ &= 2\underbrace{\cos(\theta)}_{x}\underbrace{\cos(n\theta)}_{T_n(x)} - \underbrace{\cos((n-1)\theta)}_{T_{n-1}(x)} \\ &= T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \end{aligned}$$

Nodos de Chebyshev

Como los nodos de Chebyshev están contenidos en el intervalo $\tilde{x}_k \in [-1, 1]$, para obtener los nodos en un intervalo arbitrario $x_k \in [a, b]$ usamos la transformación

$$x_k = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos\left(\frac{2k-1}{2n}\pi\right).$$





Nodos de Chebyshev

Teorema: Si $P_n(x)$ es un polinomio mónico[a] de grado n, entonces

$$\| P_n(x) \|_\infty = \max_{-1 \le x \le 1} |P_n(x)| \ge \frac{\| T_n(x) \|}{2^{n-1}} = \frac{1}{2^{n-1}}$$

Demostración: Supongamos que existe un polinomio mónico $P_n(x)$ de grado n tal que

$$|P_n(x)| \le \frac{1}{2^{n-1}} \quad , \quad x \in [-1, 1] \quad , \text{ definamos}$$

$$Q(x) = (T_n(x))/(2^{n-1}) - P_n(x)$$

Como $(T_n(x))/(2^{n-1})$ y $P_n(x)$ son mónicos de grado n entonce el grado de $Q(x)$ es $(n-1)$. En los puntos extremos de $T_n(x)$

$$x_k = \cos\left(\frac{k\pi}{n}\right) \quad \text{tenemos que} \quad T_n(x_k) = (-1)^k$$

[a] Un polinomio mónico es aquel cuyo término de mayor grado $a_n x^n$ tiene por coeficiente $a_n = 1$

Nodos de Chebyshev

Entonces

$$Q(x_k) = \frac{T_n(x_k)}{2^{n-1}} - P_n(x_k) = \frac{(-1)^k}{2^{n-1}} - P_n(x_k)$$

y como $P_n(x_k) < 1/2^{n-1}$ tenemos que

$$Q(x_k) \begin{cases} \leq 0 & \text{, si } k \text{ es impar} \\ \geq 0 & \text{, si } k \text{ es par} \end{cases} \text{, cambia de signo al menos } (n+1) \text{ veces,}$$

es decir, que $Q(x)$ tiene al menos n raices. Pero como $Q(x)$ es a lo sumo de grado $(n-1)$ entonces tenemos que

$$P_n(x_k) \equiv \frac{T_n(x_k)}{2^{n-1}}$$

Este resultado además nos permite acotar el error de interpolación

Error del polinomio interpolatorio

Suponiendo que el intervalo de interpolación sea el $I = [-1, 1]$, con $x \in I$, el error

$$\varepsilon(x) = |f(x) - P_n(x)| = |f^{n+1}(\xi)|\frac{|\phi(x)|}{(n+1)!} \leq \max|f^{n+1}(x)|\frac{\max|\phi(x)|}{(n+1)!}$$

donde, $\phi(x) = (x - x_0)\cdots(x - x_n)$ es un polinomio mónico de grado $(n + 1)$.
Si los x_i son los nodos de Chebyshev, por el teorema anterior $\max|\phi(x)| = 2^{-n}$, y en consecuencia

$$\max_{x\in I}(\varepsilon(x)) = \max_{x\in I}|f(x) - P_n(x)| = \| f(x) - P_n(x) \|_\infty \leq \frac{\| f^{n+1}(x) \|_\infty}{2^n(n+1)!}$$

Es decir, el error de interpolación usando los nodos de Chebyshev está acotado.

Spline

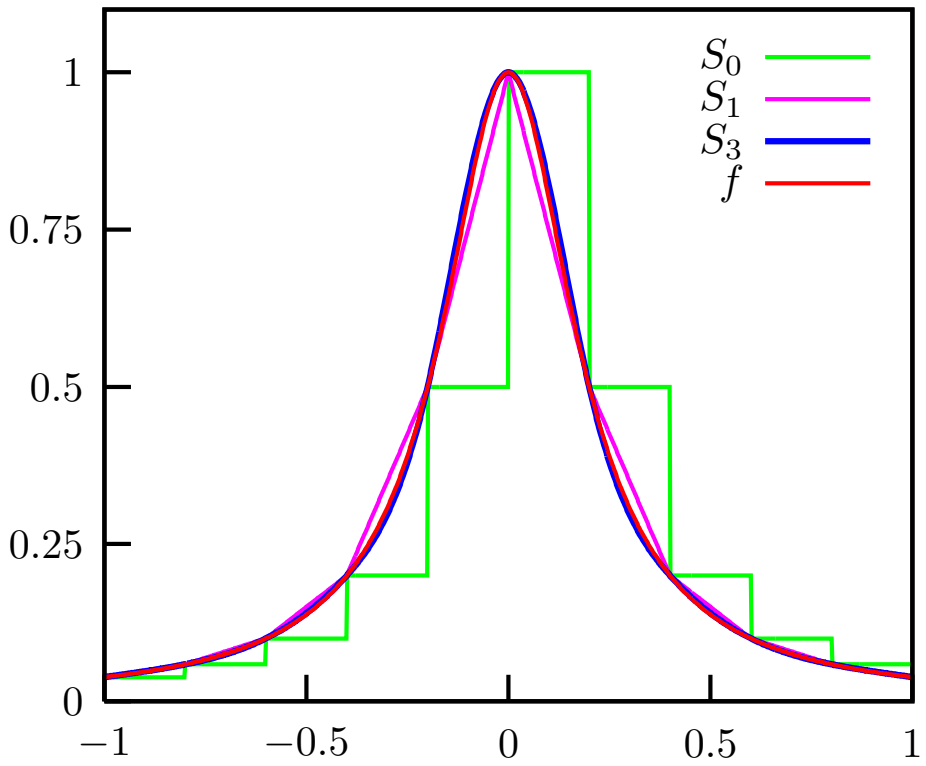
La Interpolación por Splines usa *secciones* de varios polinomios en distintos intervalos de la función a interpolar para evitar problemas de oscilación como el Fenómeno de Runge.Por ejemplo

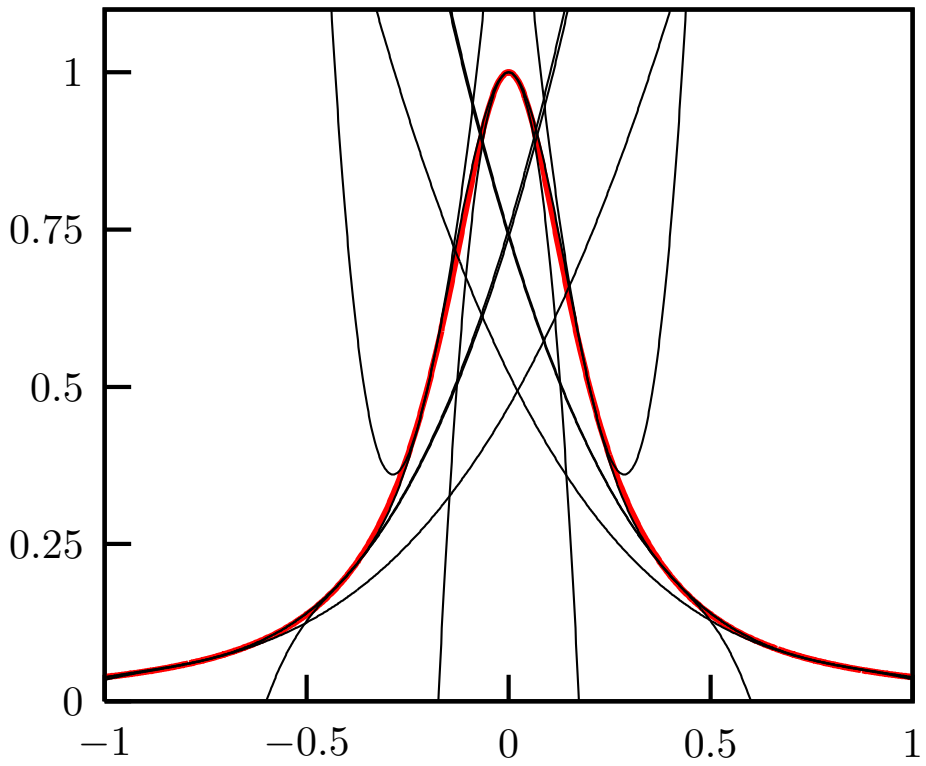
$$S_m(x) = \begin{cases} P_m^{[0]}(x) & , \text{ si } x \in [x_0, x_1] \\ P_m^{[1]}(x) & , \text{ si } x \in [x_1, x_2] \\ \vdots & \\ P_m^{[n-1]}(x) & , \text{ si } x \in [x_{n-1}, x_n] \end{cases}$$

Definición: Sea $a = x_0 < x_1 < \cdots < x_n = b$ un conjunto de puntos en $[a, b]$. La función spline o trazador de grado m con nodos en los punto x_i, con $i = 0, 1, \ldots, n$ es la función $S_m(x)$ que cumple con

- Si $x \in [x_i, x_{i+1}]$ entonces $S_m(x) = P_m^{[i]}(x)$
- Las primeras $(m-1)$ derivadas de $S_m(x)$ son continuas

Spline





Spline Cúbico

El spline cúbico $S_3(x)$, que llamaremos $S(x)$, es la más común de las interpolaciones por trazadores

Condiciones:

- $S(x) = P^{[i]}(x)$ para $x_i \leq x \leq x_{i+1}$, donde $P^{[i]}(x)$ es un pol. cúbico
- $S(x_i) = f(x_i)$ para $i = 0, 1, \ldots, n$
- $P^{[i+1]}(x_{i+1}) = P^{[i]}(x_{i+1})$ para $i = 0, 1, \ldots, n-2$
- $P^{[i+1]'}(x_{i+1}) = P^{[i]'}(x_{i+1})$ para $i = 0, 1, \ldots, n-2$
- $P^{[i]''}(x_i) = P^{[i+1]''}(x_i)$ para $i = 0, 1, \ldots, n-2$
- Y una de estas condiciones de frontera se satisface
  - $S''(x_0) = S''(x_n) = 0$ (Frontera natural o libre)
  - $S'(x_0) = f'(x_0)$ y $S'(x_n) = f'(x_n)$ (Frontera sujeta)

Spline Cúbico

Sea

$$P^{[i]}(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

y definamos $h_i = (x_{i+1} - x_i)$ para $i = 0, 1, \ldots, n-1$.
Entonces de la 2^{da} condición tenemos

$$a_i = P^{[i]}(x_i) = S(x_i) = f(x_i) \quad , \quad i = 0, 1, \ldots, n-1$$

De lo anterior y la 3^{ra} condición tenemos

$$a_{i+1} = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 \quad , \quad i = 0, 1, \ldots, n-2$$

Derivando a $P^{[i]}(x)$ y aplicanco la 4^{ta} condición tenemos

$$b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2 \quad , \quad i = 0, 1, \ldots, n-2$$

Volviendo a derivar a $P^{[i]}(x)$ y aplicando la 5^{ta} condición obtenemos

$$c_{i+1} = c_i + 3d_i h_i \quad , \quad i = 0, 1, \ldots, n-2$$

Spline Cúbico

Operando con las 4 ecuaciones previas obtenemos

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1})$$

con $i = 1, 2, \ldots, n-1$.
Como los a_i y los h_i están determinados por los $(n+1)$ puntos, nos queda un sistema de $(n+1)$ incognitas, c_i, con $(n-1)$ ecuaciones.
Las dos ecuaciones restantes las obtenemos aplicando alguna condición de frontera de tal forma que el problema se reduzca a un sistema de ecuaciones lineales de la forma

$$\mathbf{Mc} = \mathbf{z} \quad ,$$

donde, **M** es una matriz tridiagonal $(n+1) \times (n+1)$, **c** es el vector que contiene a los coeficientes c_i y **z** es un vector columna.
Conocidos todos los a_i y los c_i usando las ecuaciones anteriores se pueden calcular los b_i y los d_i

Spline Cúbico

Si enumeramos las $(n+1)$ filas de **M** desde la 0 hasta la n, los términos de **M** vienen dados por

$$m_{ij} = \begin{cases} h_{i-1} & , \text{ si } i = j-1 \text{ y } i = 1,2,\ldots,n-1 \\ 2(h_{i-1}+h_i) & , \text{ si } i = j \text{ y } i = 1,2,\ldots,n-1 \\ h_i & , \text{ si } i = j+1 \text{ y } i = 1,2,\ldots,n-1 \\ \text{depende del borde} & , \text{ para } (i,j) = (0,0);(0,1);(n,n-1);(n,n) \\ 0 & , \text{ en otros casos} \end{cases}$$

y los términos del vector **z** vienen dados por

$$z_i = \begin{cases} \frac{3}{h_i}(a_{i+1}-a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1}) & , \text{ si } i = 1,2,\ldots,n-1 \\ \text{depende del borde} & , \text{ si } i = 0 \text{ o } i = n \end{cases}$$

Spline Cúbico

Borde	Natural	Sujeto
m_{00}	1	$2h_0$
m_{01}	0	h_0
m_{nn-1}	0	h_{n-1}
m_{nn}	1	$2h_{n-1}$
z_0	0	$\frac{3}{h_0}(a_1 - a_0) - 3f'(x_0)$
z_n	0	$3f'(x_n) - \frac{3}{h_{n-1}}(a_n - a_{n-1})$

- Como la matriz **M** es estrictamente diagonal dominante, entonces en no singular, por lo que el sistema de ecuaciones lineales tiene una solución y esta es única
- Se pueden establecer otras condiciones de borde pero esto implica un cálculo de todos los términos de las ecuaciones 0 y n del sistema lo que puede traer como consecuencia la perdida de la forma tridiagonal de **M**

Polinomio de Interpolación

Comentarios

- No se mejoran los resultados tomando un numero elevado de puntos en un mismo intervalo, pues veremos que esto acarrea mejoras en determinadas zonas pero notables mermas de precision en otras, fenomeno Runge: el error de interpolacion es menor en la zona central del intervalo y mayor en los extremos.
- Desde el punto de vista numerico es preferible, en vez de generar un único polinomio interpolador en base a muchos puntos de un intervalo, dividir el intervalo en otros de modo que por medio de varios polinomios lograr mejorar la precision en la interpolacion.
- Debemos seleccionar un intervalo de interpolacion tal que el punto que queremos aproximar se encuentre en la zona central del soporte.
- No debemos usar el polinomio para aproximar valores que esten fuera del intervalo de interpolacion.

Integración Numérica

- Fórmulas de integración basadas en interpolación.
  - Análisis del error
- Cuadratura gaussiana.
- Método de Monte Carlos

Integración basada en interpolación

El problema a resolver numéricamente es

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

A diferencia del caso analítico, el problema numérico es mucho más sencillo tanto así que numéricamente podemos decir:

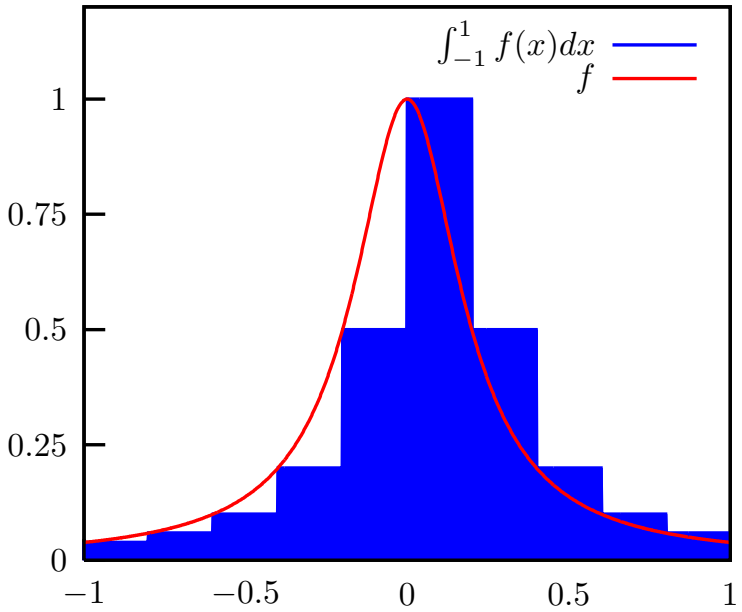
Deriva el que puede e integra el que sabe

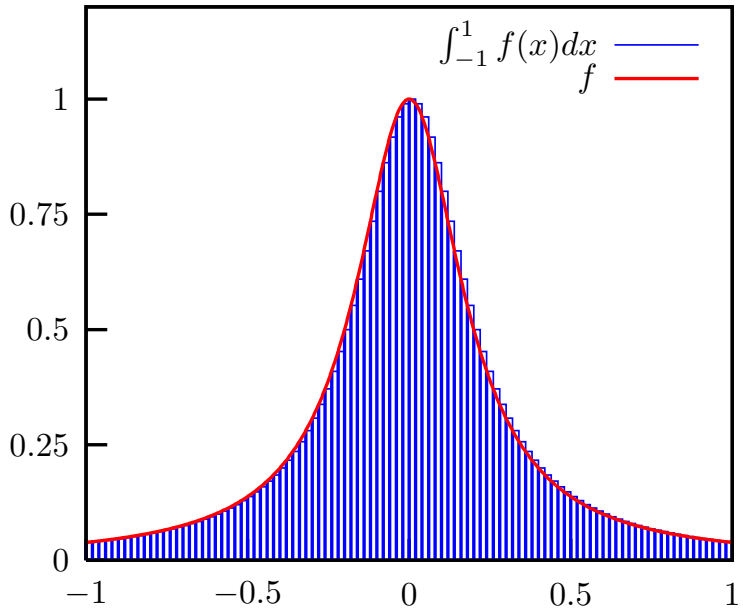
Definición de la integral Riemman

$$I = \int_a^b f(x)dx = \lim_{h \to 0} \left(h \sum_{i=0}^{\frac{b-a}{h}-1} f(x_i) \right) \quad \text{, con } x_i = a + ih$$

En general, la idea es aproximar la integral por el área bajo la curva medida en cuadrículas

Integración basada en interpolación





La expresión general para la integración numérica es

$$I = \int_a^b f(x)dx \approx \sum_{i=0}^{n-1} f(x_i)w_i \quad , \quad a \leq x_i \leq b \quad \forall \quad i$$

donde w_i son los pesos asociados al método.

Integración basada en interpolación

Uno de los esquemas es el “cerrado” de Newton-Cotes para funciones razonablemente bien comportadas. Los puntos, x_i se toman igualmente espaciados en el intervalo $[a, b]$ con $x_0 = a$ y $x_n = b$, así tenemos

$$x_{i+1} = x_i + h = x_0 + ih \quad , \text{ con } h = \text{ctte y } n = (b-a)/h$$

Si expandimos a $f(x)$ en series de potencias obtenemos

$$I = \sum_{i=0}^{n-1} \int_{x_i}^{x_i+h} \left(f(x_i) + xf'(x_i) + \frac{x^2}{2!} f''(x_i) + \frac{x^3}{3!} f'''(x_i) + \cdots \right) dx$$

Usando los dos primeros términos tendremos una aproximación la función con rectas y la expresión anterior queda

$$\sum_{i=0}^{n-1} \int_{x_i}^{x_i+h} \left(f(x_i) + xf'(x_i) + O(x^2) \right) dx$$

Método de los trapecios

Una forma de trazar las rectas, si se desconoce la derivada de la función, es mediante el polinomio de interpolación de primer orden. Luego para calcular numéricamente la integral calculamos el área bajo los segmentos de recta, es decir las áreas de los trapecios.

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} \left(\frac{x - x_{i+1}}{x_i - x_{i+1}} f(x_i) + \frac{x - x_i}{x_{i+1} - x_i} f(x_{i+1}) \right) dx$$

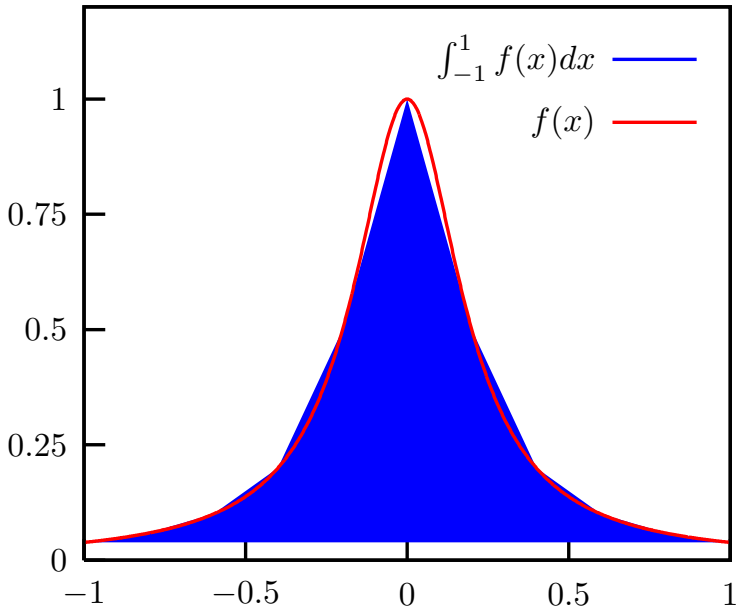
$$= \sum_{i=0}^{n-1} \left(h \left(\frac{1}{2} f(x_i) + \frac{1}{2} f(x_{i+1}) \right) \right)$$

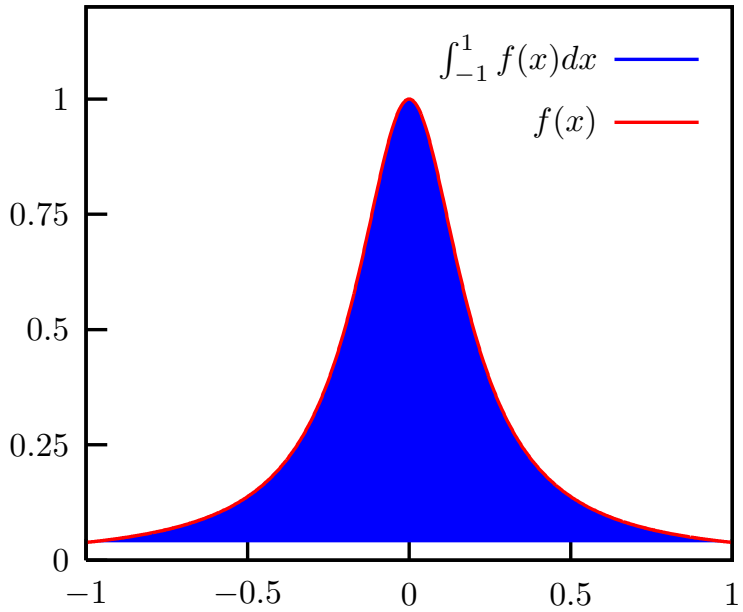
$$= \frac{h}{2} f(x_0) + h f(x_i) + h f(x_2) + \cdots + h f(x_{n-2}) + h f(x_{n-1}) + \frac{h}{2} f(x_n) \quad ,$$

donde, los pesos w_i vienen dados por

$$w = \frac{h}{2}, h, h, \ldots, h, h, \frac{h}{2}$$

Método de los trapecios





Para calcular el error del método definamos

$$I_i = \int_{x_i}^{x_{i+1}} f(x)dx \quad , \quad A_i = h\left(\frac{1}{2}f(x_i) + \frac{1}{2}f(x_{i+1})\right)$$ y al error como

$$E_i = |A_i - I_i|$$

Método de los trapecios

Expandiendo a f(x) alrededor de x_i para calcular x_{i+1} tenemos

$$f(x_{i+1}) = f(x_i) + (\overbrace{x_{i+1} - x_i}^{h}) f'(x_i) + \frac{(\overbrace{x_{i+1} - x_i}^{h})^2}{2!} f''(x_i) + \cdots$$

Entonces

$$\begin{aligned} A_i &= \frac{h}{2}\left(f(x_i) + f(x_{i+1})\right) = \frac{h}{2}\left(f(x_i) + f(x_i) + hf'(x_i) + h^2 f''(x_i) + \cdots\right) \\ &= hf(x_i) + \frac{h^2}{2} f'(x_i) + \frac{h^3}{2\times 2!} f''(x_i) + \cdots \end{aligned}$$

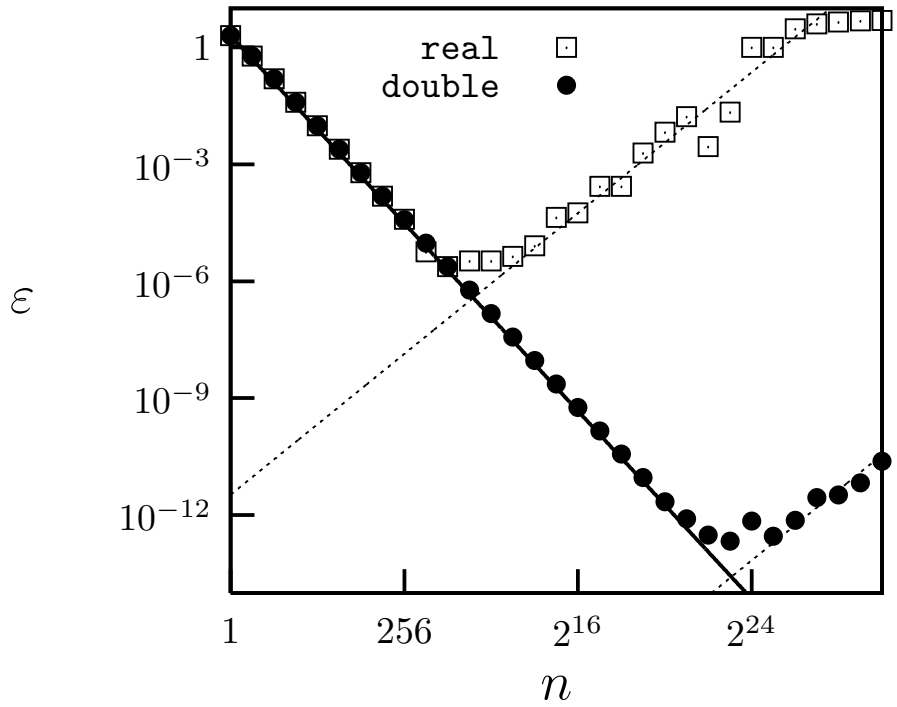
por otro lado

$$\begin{aligned} I_i &= -F(x_i) + F(x_i + 1) = -F(x_i) + \left(F(x_i) + hF'(x_i) + \frac{h^2}{2!} F''(x_i) + \cdots\right) \\ &= h\overbrace{F'(x_i)}^{f(x_i)} + \frac{h^2}{2!}\overbrace{F''(x_i)}^{f'(x_i)} + \frac{h^3}{3!}\overbrace{F'''(x_i)}^{f''(x_i)} + \cdots \end{aligned}$$

Método de los trapecios

Si tenemos que

$$\varepsilon_i = |A_i - I_i| = \frac{h^3}{2 \times 2!} f''(x_i) - \frac{h^3}{3!} f''(x_i) + O(h^4) \quad \Longrightarrow \quad \varepsilon_i \approx \frac{1}{12} h^3 f''(x_i)$$



$$\int_0^1 6x - x^6 dx = 5$$

$$\varepsilon = \varepsilon_t + \varepsilon_{maq}$$

$$\varepsilon_t \approx h^2$$

$$\varepsilon_{maq} \approx h^{-2/3}$$

Método de Simpson

Si extendemos la aproximación de la función hasta el término cuadrático tenemos

$$I = \int_a^b f(x)dx = \sum_{i=0}^{n-2} \int_{x_i}^{x_i+2h} \left(f(x_i) + xf'(x_i) + \frac{1}{2!}x^2 f''(x_i) + O(x^3) \right) dx$$

donde, $h = (b-a)/2n$. Esto equivale a aproximar a $f(x)$ usando un polinomio de grado 2 en cada intervalo $[x_i, x_{i+2}]$.

$$I \approx \sum_{i=1}^{n-2} \int_{x_i}^{x_{i+2}} \left(L_i f(x_i) + L_{i+1} f(x_{i+1}) + L_{i+2} f(x_{i+2}) \right) dx$$

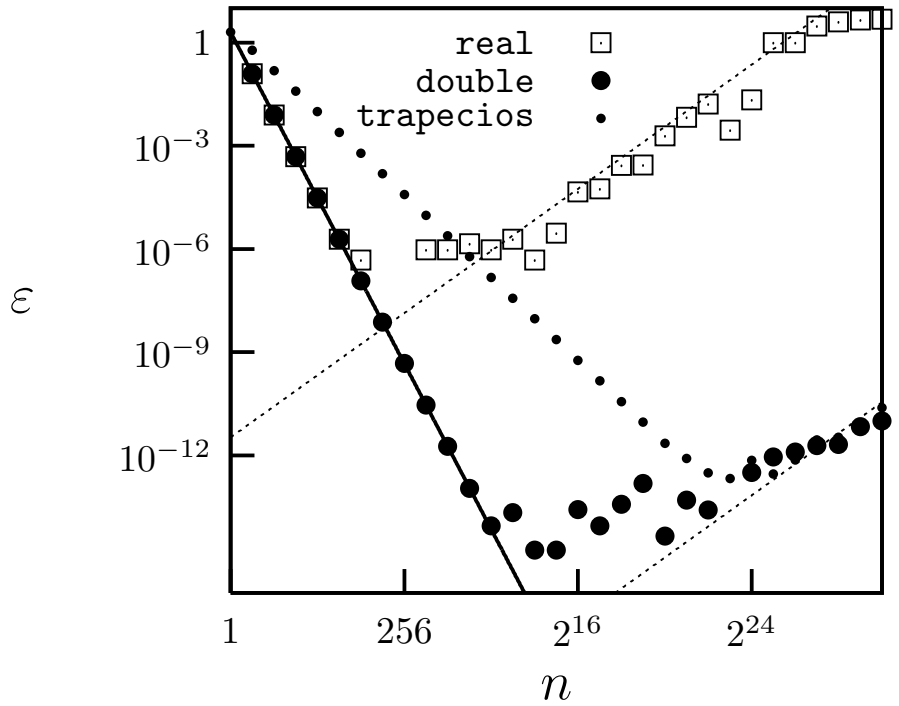
donde , $L_i = \frac{(x-x_{i+1})(x-x_{i+2})}{(x_i-x_{i+1})(x_i-x_{i+2})}$ es un polinomio de Lagrange

$$= \sum_{i=0}^{n-2} \left(h \left(\frac{1}{3} f(x_i) + \frac{4}{3} f(x_{i+1}) + \frac{1}{3} f(x_{i+2}) \right) \right)$$

Método de Simpson

El número de divisiones debe ser par y los pesos w_i seran

$$w = \left\{ \frac{1}{3}h, \frac{4}{3}h, \frac{2}{3}h, \frac{4}{3}h, \frac{2}{3}h, \ldots, \frac{4}{3}h, \frac{2}{3}h, \frac{1}{3}h \right\}$$



$$\int_0^1 6x - x^6 dx = 5$$

$$\varepsilon = \varepsilon_t + \varepsilon_{maq}$$

$$\varepsilon_t \approx h^4$$

$$\varepsilon_{maq} \approx h^{-2/3}$$

Integración basada en interpolación

Los pesos de los métodos integración usando polinomios de interpolación son

Método	Ordel	Pesos en $[x_i, x_{i+1}]$
Trapecios	1	$\left\{\frac{h}{2}, \frac{h}{2}\right\}$
Simpson	2	$\left\{\frac{h}{3}, \frac{4h}{3}, \frac{h}{3}\right\}$
3/8	3	$\left\{\frac{3h}{8}, \frac{9h}{8}, \frac{9h}{8}, \frac{3h}{8}\right\}$
Milne	4	$\left\{\frac{14h}{45}, \frac{64h}{45}, \frac{24h}{45}, \frac{64h}{45}, \frac{14h}{45}\right\}$

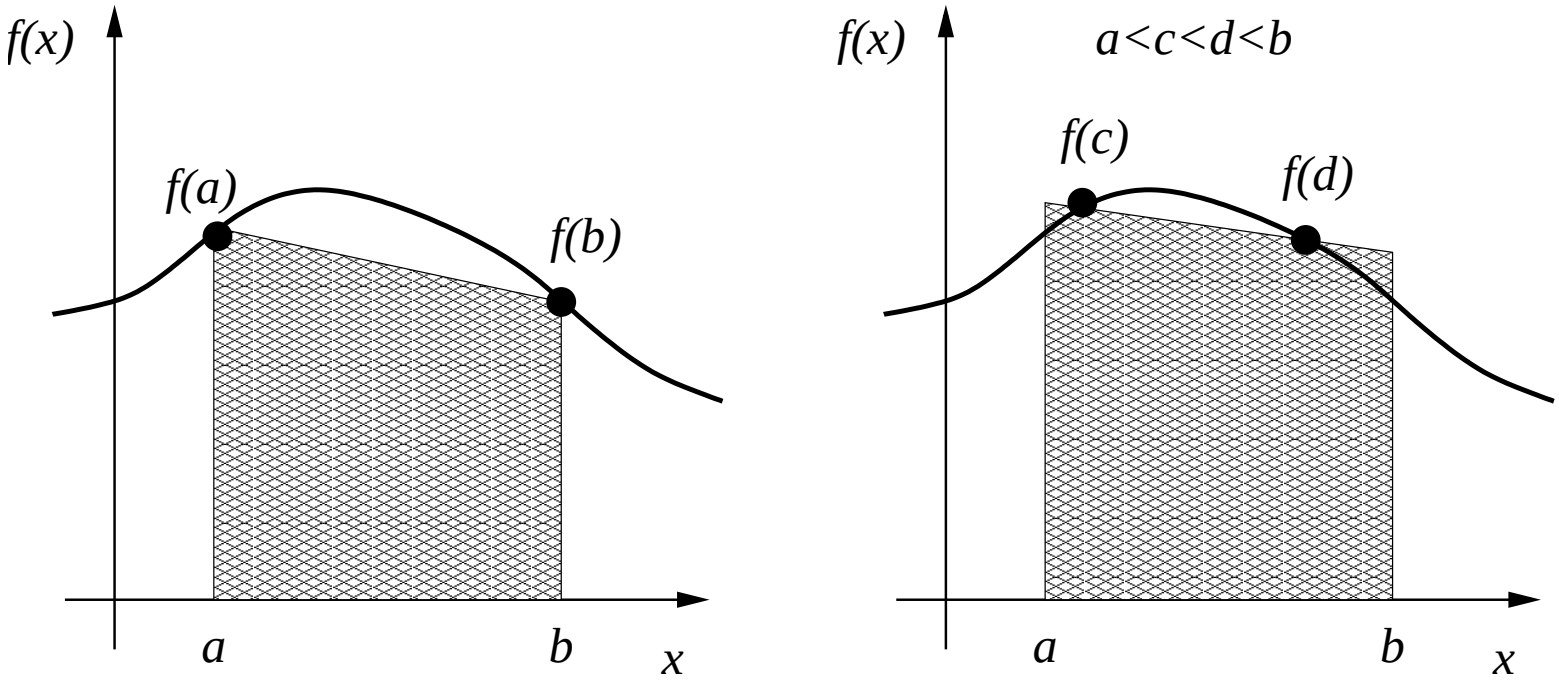
Si contamos con $n+1$ puntos en $[a, b]$ y usamos un método de orden $p \leq n$, podemos hallar los pesos w, tales que

$$\int_a^b f(x)dx = \sum_{i=1}^{n} w_i f(x_i) \quad ,$$

siempre que $f(x)$ sea de grado $\leq p$

Cuadratura Gaussiana

Gauss determinó que la selección de puntos equidistantes en los métodos de Newton-Cotes y Romberg limita su exactitud



$$I = \int_a^b f(x)dx \approx w_c f(c) + w_d f(d)$$

ahora tenemos $2n$ incongnitas

Cuadratura Gaussiana

Podemos obtener resultados exactos al integrar funciones de grado $(2n-1)$, mediante la selección de n puntos x_i y n pesos w_i. En el intervalo $[-1,1]$, la integral vienen dada por

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i) \quad , \quad w_i = \int_{-1}^{1} \prod_{j\neq i} \frac{x-x_j}{x_i-x_j} dx \quad ;$$

donde, los n puntos x_i son las raices de un polinomio perteneciente a la familia de polinomios “ortogonales” de Legendre.

$$P_0(x) = 1 \quad , \quad P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} \left[(x^2-1)^n\right].$$

Para cambiar el intervalo de la integral podemos usar la expresión

$$\begin{aligned} \int_a^b f(t)\,dt &= \tfrac{b-a}{2} \int_{-1}^{1} f\left(\tfrac{b-a}{2}x + \tfrac{a+b}{2}\right) dx \Rightarrow \\ &\approx \tfrac{b-a}{2} \sum_{i=1}^{n} w_i f\left(\tfrac{b-a}{2}x_i + \tfrac{a+b}{2}\right) \end{aligned}$$

Cuadratura Gaussiana

Los valores de x_i y w_i de cuadratura Gaussiana para valores pequeños de n son

Número de Puntos, n	Puntos, x_i	Pesos, w_i
1	0	2
2	$\pm\sqrt{1/3}$	1
3	0	$8/9$
	$\pm\sqrt{3/5}$	$5/9$
4	± 0.339981044	0.652145155
	± 0.861136312	0.347854845
5	0	0.568889
	± 0.538469	0.478629
	± 0.906180	0.236927

Cuadratura Gaussiana

El problema de la cuadratura Gaussiana se puede expresar de forma más general si introducimos una función de peso $W(x)$

$$\int_a^b W(x)\, f(x)\, dx \approx \sum_{i=1}^{n} w_i f(x_i) \quad ,$$

donde, si $a = -1$, $b = 1$, y $W(x) = 1$, tenemos el caso anterior.

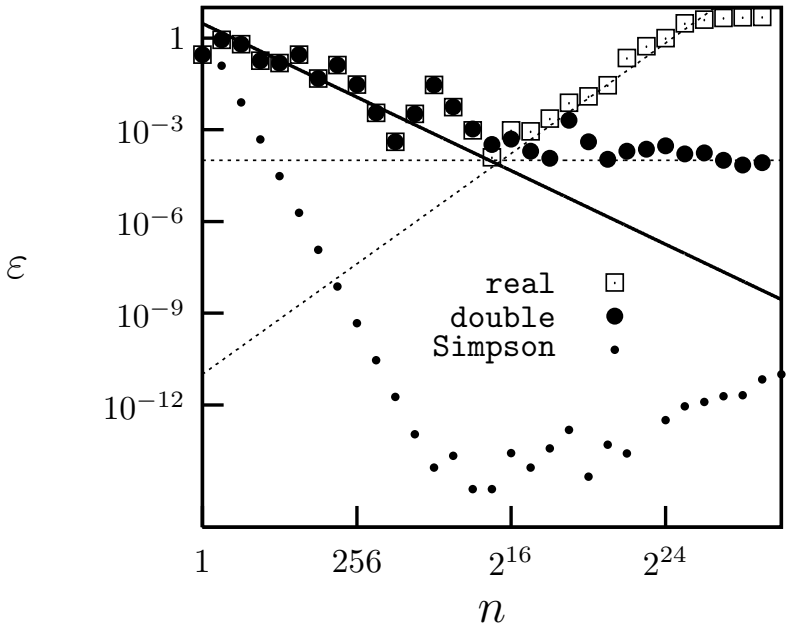
Intervalo $[a, b]$	Peso $W(x)$	Familia de Polinomios
$[-1, 1]$	1	Legendre
$(-1, 1)$	$(1-x)^\alpha (1+x)^\beta, \quad \alpha, \beta > -1$	Jacobi
$(-1, 1)$	$\frac{1}{\sqrt{1-x^2}}$	Chebyshev de primer tipo
$[-1, 1]$	$\sqrt{1-x^2}$	Chebyshev de segundo tipo
$[0, \infty)$	e^{-x}	Laguerre
$(-\infty, \infty)$	e^{-x^2}	Hermite

Método de Monte Carlo

Se basa en el teorema del valor medio para las integrales

$$I = \int_a^b dx f(x) = (b-a) f(\xi) \approx \underbrace{\frac{(b-a)}{n}}_{w=ctte} \sum_{i=1}^{n} f(x_i) \quad , \quad \xi \in [a,b]$$

donde, los n puntos x_i son seleccionados aleatoriamente.



$$\int_0^1 6x - x^6 dx = 5$$

$$\varepsilon = \varepsilon_t + \varepsilon_{maq}$$

$$\varepsilon_t \approx h^1$$

$$\varepsilon_{maq} \approx h^{-2/3}$$