Contents

Introduccion

Preliminares y definiciones basicas

Problema numerico

Un problema numérico es una clara y nada ambigua descripción de la conexión funcional entre datos de entrada (variables independientes del problema o input) y los datos de salida (resultados deseados o output).

Estos datos, input y output, consisten en un conjunto finito de cantidades reales.

Algoritmo

Un algoritmo para un problema numérico es una completa descripción de un número finito de pasos con operaciones bien definidas, y sin ambigüedades, a través de las cuales una lista de datos de entrada se convierte en una lista de datos de salida.

Preliminares matematicos

Valor intermedio para funciones continuas

Sea f continua en $[a, b]$. Sea d entre $f(a)$ y $f(b)$ entonces existe $c \in [a, b]$ tal que $f(c) = d$

Valor medio

Sea f continua en $[a, b]$ y derivable en (a, b). Entonces para todo par $x, c \in [a, b]$ se cumple que:

$\dfrac{f(x) - f(c)}{x - c} = f'(\epsilon)$ para algun ϵ entre x y c

Esto dice que $f(x) = f(c) + f'(\epsilon)(x - c)$

Taylor

Si $f \in C^{(n)}[a, b]$ y existe $f^{(n+1)}(a, b)$ entonces para todo par $x, c \in [a, b]$ se tiene que:

$$f(x) = \sum_{k=0}^{n} \frac{1}{k!} f^{(k)}(c)(x - c)^k + E_n(x),$$

donde

$E_n(x) = \dfrac{1}{(n+1)!} f^{(n+1)}(\epsilon)(x - c)^{n+1}$ para algun ϵ entre x y c

Observacion: tomando $y = c, (x - c) = h$ y por lo tanto $x = y + h$, entonces

$$f(y+h) = f(y) + hf'(y) + \frac{h^2}{2}f''(y) + \cdots + \frac{h^n}{n!}f^n(y) + \frac{h^n+1}{(n+1)!}f^{(n+1)}(\epsilon)$$

para algun ϵ entre y e $(y+h)$

Taylor con resto integral

Si $f \in C^{(n)}[a,b]$ y existe $f^{(n+1)}(a,b)$ entonces para todo par $x, c \in [a,b]$ se tiene que:

$$f(x) = \sum_{k=0}^{n} \frac{1}{k!} f^{(k)}(c)(x-c)^k + R_n(x),$$

donde

$$R_n(x) = \frac{1}{n!} \int_c^x f^{(n+1)}(t)(x-t)^n \ dt$$

Tasa de convergencia

Lineal

Sea x_n una sucesion de numeros reales que converge a x_*

Se dice que la sucesion x_n tiene tasa de convergencia (al menos) lineal si existe una constante c tal que $0 < c < 1$ y un $N \in \mathbb{N}$ tal que

$|x_{n+1} - x_*| \leq c|x_n - x_*|$ para todo $n \geq N$

Superlineal

Sea x_n una sucesion de numeros reales que converge a x_* Se dice que la tasa de convergencia es (al menos) **superlineal** si existe una sucesion ϵ_n que converge a 0 y un $N \in \mathbb{N}$ tal que

$|x_{n+1} - x_*| \leq \epsilon_n |x_n - x_*|$ para todo $n \geq N$

Cuadratica

Sea x_n una sucesion de numeros reales que converge a x_* Se dice que la tasa de convergencia es (al menos) **cuadratica** si existe una constante positiva c y un $N \in \mathbb{N}$ tal que

$|x_{n+1} - x_*| \leq c|x_n - x_*|^2$ para todo $n \geq N$

Notacion O grande y o chica

O grande

Sean $\{x_n\}$ y $\{\alpha_n\}$ dos sucesiones distintas. Se dice que

$x_n = O(\alpha_n)$

si existen una constante $C > 0$ y $r \in \mathbb{N}$ tal que $|x_n| \leq C|\alpha_n|$ para todo $n \geq r$

O chica

Se dice que

$x_n = o(\alpha_n)$

si existe una sucesion $\{\epsilon_n\}$ que converge a 0, con $\epsilon_n \geq 0$ y un $r \in \mathbb{N}$ tal que $|x_n| \leq \epsilon_n|\alpha_n|$ para todo $n \geq r$.

Intuitivamente esto dice que $\lim\limits_{n\to\infty} \left(\frac{x_n}{\alpha_n}\right) = 0$

Comparar funciones

O grande

Esta notacion tambien se puede usar para comparar funciones. Se dice que

$f(x) = O((g(x))$ cuando $x \to \infty$

si existen una constante $C > 0$ y $r \in \mathbb{R}$ tal que $|f(x)| \leq C|g(x)|$ para todo $x \geq r$.

O chica

Analogamente, se dice que $f(x) = o(g(x))$ cuando $x \to \infty$ si $\lim\limits_{n\to\infty} \left(\frac{f(x)}{g(x)}\right) = 0$

Comparar funciones cuando $\mathbf{x} \to \mathbf{x}_*$

O grande

Se dice que

$f(x) = O(g(x))$ cuando $x \to x_*$

si existen una constante $C > 0$ y un entorno alrededor de x_* tal que $|f(x)| \leq C|g(x)|$ para todo x en ese entorno

O chica

Analogamente, se dice que $f(x) = o(g(x))$ cuando $x \to x_*$ si $\lim\limits_{x\to x_*} \left(\frac{f(x)}{g(x)}\right) = 0$

Algoritmo de multiplicacion encajada (Horner)

Idea

Consiste en reescribir convenientemente el polinomio $p(x)$ de modo de reducir el numero de productos

$p(x) = 2 + x(4 + x(-5 + x(2 + x(-6 + x(8 + x \cdot 10)))))$

Si el grado de $p(x)$ es n, se requieren n productos

Descripcion matematica

Si $p(x) = a_0 + a_1 x + \cdots + a_n x^n$ con $a_n \neq 0$, la evaluacion de $p(x)$ en $x = z$ se realiza con los siguientes pasos:

$$
\begin{aligned}
b_{n-1} &= a_n \\
b_{n-2} &= a_{n-1} + z \cdot b_{n-1} \\
&\vdots \\
b_0 &= a_1 + z \cdot b_1 \\
p(z) &= a_0 + z \cdot b_0
\end{aligned}
$$

Pseudocodigo

Dados el polinomio $p(x)$, de grado n, con coeficientes a_i, para $i = 0, \ldots, n$ con $a_n \neq 0$ y un numero real z en el que se desea evaluar $p(x)$

input n; a_i, $i = 0, ..., n$; z

$b_{n-1} \leftarrow a_n$ (Asignacion)

for $k = n - 1$ **to** 0 **step** -1, **do**

$b_{k-1} \leftarrow a_k + z * b_k$

end do

output $b_i, i = -1, ..., n - 1$

Teoria de errores

Principales fuentes de error

Errores en los datos de entrada

Datos experimentales con posibles errores debido al equipamiento utilizado.

Tambien estan los errores que surgen al representar un numero real irracional con un numero finito de digitos.

Errores de redondeo

Aparecen cuando los calculos se realizan usando un numero finito de digitos.

Errores de truncamiento

Aparecen cuando un proceso finito es reemplazado por un proceso finito.

Por ejemplo: Aproximar por suma parcial, polinomio de taylor, cociente incremental, etc.

Errores humanos

Errores en la formulacion del problema, en los calculos "a mano", al escribir un programa, etc.

Errores absolutos y relativos

Concepto general

Cuando un numero real r (valor exacto) es aproximado por otro numero $\bar{r}$, se define el **error** por $r - \bar{r}$

Error absoluto

$\Delta r = |r - \bar{r}|$

Error relativo

$\delta r = |\frac{r - \bar{r}}{r}| = \frac{\Delta r}{|r|}$

Error relativo porcentual

$100 * \delta r$

Redondeo y truncado

Redondeo

Para la aproximacion por redeondeo de un numero de n digitos decimales:

- digito $(n + 1) < 5 \Rightarrow$ digito n queda igual
- digito $(n + 1) \geq 5 \Rightarrow$ se le suma 1 al digito n

Redondeo Error

Se cumple que: $|r - \bar{r}| \leq \frac{1}{2}10^{-n}$

Ejemplo:

$r = 0.11 \Rightarrow \bar{r} = 0.1$ y $|r - \bar{r}| = 0.01 \leq 0.05 = 5 \cdot 10^{-2} = \frac{1}{2}10^{-1}$

Truncado

Para la aproximacion por truncamiento de un numero de n digitos lo que se hace es cortar al numero en el digito n.

Truncado error

Se cumple que:

$|r - \bar{r}| \leq 10^{-n}$

Ejemplo:

$r = 0.11 \Rightarrow \bar{r} = 0.1$ y $|r - \bar{r}| = 0.01 \leq 0.1 = 10^{-1}$

Digitos significativos

El numero $\bar{r}$ se aproxima a r con m **digitos significativos** si

$\delta r = \dfrac{\Delta r}{|r|} \leq 5 \cdot 10^{-m}$

Esto dice que el error relativo es del orden de 10^{-m}

Erorres en las operaciones

Suma

Sean $x_1, x_2 \in \mathbb{R}$ y $\bar{x_1}, \bar{x_2}$ aproximaciones de x_1 y x_2 respectivamente

Sean $y = x_1 + x_2, \quad \bar{y} = \bar{x_1} + \bar{x_2}$

El error en la operacion **suma** está dado por:

$y - \bar{y} = (x_1 + x_2) - (\bar{x_1} + \bar{x_2}) = (x_1 - \bar{x_1}) + (x_2 - \bar{x_2})$

Error absoluto $\Delta y = |y - \bar{y}| \leq |x_1 - \bar{x_1}| + |x_2 - \bar{x_2}$

$\Delta y \leq \Delta x_1 + \Delta x_2$

Error relativo $\delta y = \dfrac{\Delta y}{|y|} \leq \dfrac{\Delta x_1 + \Delta x_2}{|x_1 + x_2}$

si $y = \displaystyle\sum_{i=1}^{n} x_i \Rightarrow \Delta y \leq \sum_{i=1}^{n} \Delta x_i$

Resta

Error absoluto Sean $y = x_1 + x_2, \quad \bar{y} = \bar{x_1} + \bar{x_2}$

$\Delta y = |y - \bar{y}| = |(x_1 - x_2) - (\bar{x_1} - \bar{x_2})| = |(x_1 - \bar{x_1})| - (x_2 - \bar{x_2})|$

$\Delta y \leq \Delta x_1 + \Delta x_2$

Error relativo Sean $y = x_1 + x_2, \quad \bar{y} = \bar{x_1} + \bar{x_2}$

$\delta y = \dfrac{\Delta y}{|y|} \leq \dfrac{\Delta x_1 + \Delta x_2}{|x_1 - x_2|}$

Multiplicacion y division

Definimos $y = x_1 * x_2, \bar{y} = \bar{x_1} * \bar{x_2}, z = \dfrac{x_1}{x_2}, \bar{z} = \bar{x_1} * \bar{x_2}$

Se puede deducir que:

$$\Delta y \lessapprox |x_2|\Delta x_1 + |x_1|\Delta x_2 \qquad \delta y = \frac{\Delta y}{|y|} \lessapprox \frac{\Delta x_1}{|x_1|} + \frac{\Delta x_2}{|x_2|}$$

y que:

$$\Delta z \lessapprox \frac{1}{|x_2|}\Delta x_1 + \frac{|x_1|}{|x_2^2|}\Delta x_2 \qquad \delta z = \frac{\Delta z}{|z|} \lessapprox \frac{\Delta x_1}{|x_1|} + \frac{\Delta x_2}{|x_2|}$$

Cancelacion de digitos significativos

Ejemplo

Sean $x_1 = 10.123455 \pm 0.510^{-6}, \quad x_2 = 10.123789 \pm 0.510^{-6}$

x_1 y x_2 ambos tienen 8 digitos significativos.

La resta $y = x_1 - x_2 = -0.000334 \pm 10^{-6}$ tiene un error relativo:

$\dfrac{\Delta y}{|y|} \leq \dfrac{10^{-6}}{0.000334} < 3\ 10^{-3} < 5\ 10^{-3}$,

Por lo tanto la resta tiene solo 3 digitos significativos

Por lo cual conviene evitar restas de numeros proximos siempre que sea posible.

Representacion de numeros en una computadora

Observaciones

Representacion en cualquier base La mayoria de los numeros reales no pueden ser representados exactamente en cualquier base.

Errores al pasar de base Aparecen errores de representacion cuando un numero es convertido de un sistema de numeracion a otro

Errores por aritmetica finita Aparecen errores debido a que la computadora usa aritmetica finita

Sistema de punto fijo

Se representan usando una cantidad fija de numeros enteros y numeros fraccinarios

La desventaja de este sistema es que no es posible representar simultaneamente numeros reales muy pequeños y muy grandes

Sistema de punto flotante

Definicion Es el conjunto de numeros normalizados en punto flotante en el sistema de numeracion con base β y t digitos para la parte fraccionaria, es decir, numeros de la forma:

$x = m\beta^e$

donde:

$m = \pm 0.d_{-1}d_{-2}\dots d_{-t}$

con $d_{-i} \in \{0, \dots, \beta - 1\}$ para $i = 1, \dots t$, con $d_{-1} \neq 0$ y $L \leq e \leq U$

Ademas, β, e y m se denominan base, exponente y mantisa respecitvamente.

Es decir, $\frac{1}{\beta} \leq |m| < 1$

Observaciones Puede ocurrir overflow si $e > U$ o underflow si $e < L$

El cero no puede representarse en este sistema de numeros normalizados

Errores de redondeo

Sea $x = m\beta^e, \qquad \frac{1}{\beta} \leq |m| < 1,$

Donde el exponente e es tal que $L \leq e \leq U$

Su representacion como numero flotante es:

$$fl(x) = x_r = m_r\beta^e, \qquad \frac{1}{\beta} \leq |m| < 1,$$

Donde m_r es la mantisa que se obtiene redondeando a t digitos la parte fraccionaria de m.

Entonces es claro que:

$$|m_r - m| \leq \frac{1}{2}\beta^{-t},$$

Error absoluto de representacion en x es:

$$|x_r - x| \leq \frac{1}{2}\beta^{-t}\beta^{e}.$$

Para el error relativo tenemos lo siguiente:

$$\frac{|x_r - x|}{|x|} \leq \frac{\frac{1}{2}\beta^{-t}\beta^{e}}{|m|\beta^{e}} = \frac{1}{2|m|}\beta^{-1} \leq \frac{1}{2}\beta^{1-t},$$

Pues si $|m| \geq \dfrac{1}{\beta} \quad \Rightarrow \quad \dfrac{1}{|m|} \leq \beta$

Luego el error relativo está acotado por:

$$\frac{|x_r - x|}{|x|} \leq \frac{1}{2}\beta^{1-t} = \mu,$$

Donde μ se llama unidad de redondeo

Notar que el error absoluto de reprsentacion en punto flotante depende del orde de la magnitud, en cambio el error relativo no

Axiomas que no aplican a punto flotante Asociatividad:

$$fl(fl(a+b)+c) \neq fl(a+fl(b+c))$$

Observaciones de implementacion Conviene reemplazar

```
if x == y then...
```

por

```
if (abs(x-y)) < epsilon then...
```

Ya que es casi imposible que se verifique la primer sentencia

Solucion de ecuaciones no lineales

Problema

Dada $f : \mathbb{R}^n \to \mathbb{R}$ no lineal, se desea encontrar una solucion r de la ecuacion:

$$f(x) = 0$$

Idea: comenzando con algun $x_0 \in \mathbb{R}$, generar una sucesion $\{x_k\}$ a traves de un algoritmo numerico iterativo, y se espera que tal sucesion converja a r donde $f(r) = 0$

Metodo de biseccion

Existencia de raiz

Si f es continua en $[a, b]$ y si $f(a) \cdot f(b) < 0 \Rightarrow f$ debe tener una raiz en (a, b)

Idea

Si $f(a)f(b) < 0$, se calculan $c = \dfrac{a+b}{2}$ y $f(c)$

Sean

- $x_0 = c$: una aproximacion de la raiz r de f y
- $|e_0| = |x_0 - r| \leq \dfrac{b-a}{2}$: error de aproximacion inicial

Se tienen 3 posibilidades:

1) Si $f(a)f(c) < 0$ entonces hay una raiz en el intervalo $[a, c]$. Reasignamos $b \leftarrow c$ y se repite el procedimiento en el nuevoo intervalo $[a, b]$
2) Si $f(a)f(c) > 0$ entonces hay una raiz en el intervalo $[c, b]$. Reasignamos $a \leftarrow c$ y se repite el procedimiento en el nuevo intervalo $[a, b]$
3) Si $f(a)f(c) = 0$ entonces $f(c) = 0$ y $x_0 = c$ es la raiz buscada
   Esto se da rara vez en la practica por cuestiones de redondeo
   Lo que en realidad se hace es ver si $|f(c)| < TOL$, donde TOL es una tolerancia dada por el usuario

Comentarios de implementacion

Calcular $c \leftarrow \dfrac{(a+b)}{2}$

- En vez de calcular $c \leftarrow \dfrac{(a+b)}{2}$, es mas conveniente calcular $c \leftarrow a + \dfrac{(b-a)}{2}$

Determinar cambio de signo

- Para determinar el cambio de signo de la funcion en vez de analizar si $f(a)f(c) < 0$, conviene usar la funcion sign y analizar si $sign(f(a) \neq sign(f(b))$

Criterios de parada Se utilizan 3 criterios de parada en el algoritmo:

1) el numero maximo de pasos permitidos
2) El error en la variable es suficientemente pequeño (δ)
3) El valor de $|f(c)|$ es suficientemente pequeño (ϵ)

Algoritmo de biseccion

Datos de entrada: * a y b extremos del intervalo

- M el maximo numero de iteraciones
- δ la tolerancia para el error e (en la variable x)
- ϵ la tolerancia para los valores funcionales

input a,b, M, δ, ϵ
$u \leftarrow f(a)$
$v \leftarrow f(b)$
$e \leftarrow b - a$
input a,b,u, v
if $sign(u) = sign(v)$ **then** STOP
for k = 1,2, ..., M **do**
  $e \leftarrow \frac{e}{2}$
  $c \leftarrow a + e$
  $w \leftarrow f(c)$
  output k,c,w,e
  if $|e| < \delta$ **or** $|w| < \epsilon$ **then** STOP
  if $sign(w) \neq sign(u)$ **then**
    $b \leftarrow c$
    $v \leftarrow w$
  else
    $a \leftarrow c$
    $u \leftarrow w$
  fi
od

Teorema del Limite

Si $[a_0, b_0], [a_1, b_1], \ldots, [a_n, b_n], \ldots$ denotan los sucesivos intervalos en el metodo de biseccion, entonces existen los limites $\lim_{n\to\infty} a_n$ y $\lim_{n\to\infty} b_n$, son iguales y representan una raiz de f

Si $c_n = \frac{1}{2}(a_n + b_n)$ y $r = \lim_{n\to\infty} c_n$,

$\Rightarrow |r - c_n| \leq \frac{1}{2^{n+1}}(b_0 - a_0)$

Relacion elementos finales e iniciales

$b_n - a_n = \frac{b_0 - a_0}{2^n}$

Metodo de Newton

Idea

Comenzando con una aproximacion x_0 de r, la iteracion del metodo de Newton consiste en calcular

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

Dado el punto $(x_n, f(x_n))$, la idea consiste en aproximar el grafico de la funcion f por la recta tangente a f que pasa por $(x_n, f(x_n))$

Algoritmo

Datos de entrada:

- x_0 : aproximacion inicial
- M: Numero maximo de iteraciones
- δ la tolerancia para el error e (en la variable x)
- ϵ la tolerancia para los valores funcionales

input x_0, M, δ, ϵ
$v \leftarrow f(x_0)$
output 0, x_0, v
if $|v| < \epsilon$ **then** STOP
for $k = 1, 2, \ldots, M$ **do**
  $x_1 \leftarrow x_0 - \dfrac{v}{f'(x_0)}$
  $v \leftarrow f(x_1)$
output k, x_1, v
  if $|x_1 - x_0| < \delta$ **or** $|v| < \epsilon$ **then** STOP
    $x_0 \leftarrow x_1$
od

Analisis de erorres

S f'' es continua en un entorno de una raiz r de f y si $f'(r) \neq 0 \Rightarrow \exists \delta > 0$ tal que si el punto inicial x_0 satisface $|r - x_0| \leq \delta$ luego todos los puntos de la sucesion $\{x_n\}$ generados por el algoritmo satisfacen que $|r - x_N| \leq \delta \forall n$, la sucesion $\{x_n\}$ converge a r y la convergencia es cuadratica

Convergencia en convexidad

Si f'' es continua en $\mathbb{R}$, f es creciente y convexa en $\mathbb{R}$ y tiene una raiz, entonces esa raiz es unica y la iteracion de Newton convergerá a esa raiz independientemente del punto inicial x_0

Metodo de la secante

Idea

La idea del metodo de la secante consiste en reemplazar $f'(x_n)$ en la iteracion de Newton por una aproximacion dada por el cociente incremental, dado por la pendiente de la recta secante que pasa por los puntos $(x_n, f(x_n))$ y $(x_n + h, f(x_n + h)$

Iteracion

La iteracion del metodo secante consiste en:

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

es decir,

$$x_{n+1} = x_n - f(x_n)\left[\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1}}\right]$$

Algoritmo

Datos de entrada:

- a: la penultima aproximacion de r
- b: la ultima aproximacion de r
- M: numero maximo de iteraciones
- δ: la toleracion para el error e (en la variable x)
- ϵ la tolerancia para los valores funcionales

input a,b,M, δ, ϵ
$fa \leftarrow f(a)$
$fb \leftarrow f(b)$
output 0,a, fa
output 1,b, fb
for k := 2 **to** M **do**
  if $|fa| < |fb|$ **then**
    $a \leftrightarrow b; fa \leftrightarrow fb$
  fi
  $s \leftarrow (b - a)/(fb - fa)$
  $b \leftarrow a$
  $fb \leftarrow fa$
  $a \leftarrow a - fa * s$
  $fa \leftarrow f(a)$
  output k,a,fa
  if $|b - a| < \delta$ **or** $|fa| < \epsilon$ **then** STOP
od

Observaciones

- En el algoritmo los puntos a y b pueden intercambiarse para lograr que $|f(b)| \leq |f(a)|$. Esto garantiza que la sucesion {|f(x_n)|} es no creciente
- Tiene convergencia superlineal
- Dos iteraciones de metodo de la secante es mejor que una iteracion del metodo de newton

Iteracion de punto fijo

Definicion de punto fijo

Un punto fijo de una funcion g es un numero p, en el dominio de g, tal que $g(p) = p$

Teoremas

Existencia Si $g \in C[a, b]$ (es decir, g es una funcion continua en $[a, b]$) y $g(x) \in [a, b] \forall x \in [a, b] \Rightarrow \exists p \in [a, b]$ tal que $g(p) = p$

Unicidad Si ademas existe $g'(x) \forall x \in (a, b)$ y existe una constante positiva $k < 1$ tal que $|g'(x)| \leq k \forall x \in (a, b) \Rightarrow$ el punto fijo en (a, b) es unico

Convergencia al unico punto fijo Sea $g \in C[a, b]$ tal que $g(x) \in [a, b] \forall x \in [a, b]$.

Supongamos que $\exists g'(x) \forall x \in (a, b)$ y existe una constante positiva $0 < k < 1$ tal que $|g'(x)| \leq k \forall x \in (a, b) \Rightarrow$ para cualquier $p_0 \in [a, b]$ la sucesion definida por $p_n = g(p_{n-1})$ para $n \geq 1$, converge al unico punto fijo $p \in (a, b)$

Idea del algoritmo de punto fijo

Para calcular aproximadamente el punto fijo de una funcion g primero se inicia con una aproximacion lineal p_0 y calculando $p_n = g(p_{n-1})$ para $n \geq 1$ se obtiene una sucesion de aproximaciones $\{p_n\}$. Si la funcion g es continua y la sucesion converge entonces lo hace a un punto fijo p de g pues:

$$p = \lim_{n \to \infty} p_n = \lim_{n \to \infty} g(p_{n-1}) = g(\lim_{n \to \infty} p_{n-1}) = g(p)$$

Algoritmo

Datos de entrada:

- p_0 : una aproximacion inicial
- M: el numero maximo de iteraciones
- δ: la tolerancia para el error e (en la variable x)

```
input p_0, M, δ
output 0, p_0
i ← 1
while i ≤ M do
   p ← g(p_0)
   output i,p
   if |p − p_0| < δ then STOP fi
   i ← i + 1
   p_0 ← p
od
```

Analisis de error en metodos de punto fijo

Cotas de error Si g es una funcion que satisface las hipotesis del teorema teorema de convergencia al unico punto fijo, se tienen las siguientes cotas de error:

$$|p_n - p| \leq k^n \ \max\{p_0 - a, b - p_0\}$$

$$|p_n - p| \leq \frac{k^n}{1-k}|p_1 - p_0| \qquad \forall n \geq 1$$

Orden de convergencia Si las derivadas de la funcion de iteracion de punto fijo se anulan en el punto fijo p hasta el orden (r-1) entonces el metodo tiene orde de convergencia (de al menos) r

Metodo de newton como metodo de punto fijo Si f es una funcion que tiene una raiz simple p, entonces el metodo de Newton es un metodo de punto fijo y tiene orden de convergencia (de al menos) 2

Multiplicidad $r \geq 2$ de f Si p es una raiz de multiplicidad $r \geq 2$ de $f \Rightarrow$ el metodo de Newton tiene orden 1

Recuperacion de convergencia cuadratica Si p es una raiz de multiplicidad $r \geq 2$ de $f \Rightarrow$ la siguiente modificacion del metodo de Newton recupera la convergencia cuadratica

$$x_{n+1} = x_n - r\frac{f(x_n)}{f'(x_n)}, \qquad \text{esto es} \qquad g(x) = x - r\frac{f(x)}{f'(x)}$$

Interpolacion polinomial

Existencia y unicidad

Dados $x_0, \dots, x_n$ numeros reales distintos con valores asociados $y_0, \dots, y_n$, entonces existe un unico polinomio p_n de grado menor o igual a n tal que $p_n(x_i) = y_i$, para $i = 0, \dots, n$

Formas del polinomio interpolante

Forma de Newton

La forma compacta del polinomio interpolante de Newton es:

$$p_k(x) = \sum_{i=0}^{k} c_i \prod_{j=0}^{i-1} (x - x_j)$$

Se adopta la convencion de que:

$$\prod_{j=0}^{m} (x - x_j) = 1 \text{ si } m < 0$$

Para evaluar $p_k(x)$, una vez calculados los coeficientes c_k, conviene usar el algoritmo de Horner

Forma de Lagrange

Sea

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)} \qquad \text{para } i = 0, \dots, n$$

$$p_n(x) = \sum_{i=0}^{n} y_i \; l_i(x)$$

Error en el polinomio interpolante

Observaciones

Derivada (n+1) de un polinomio Si p es un polinomio de grado igual a n $\Rightarrow p^{(n+1)(x) \equiv 0}$

Teorema de rolle Si f es una funcion continua en $[a, b]$ y derivable en (a, b)

Si ademas $f(a) = f(b) \Rightarrow \exists \alpha \in (a, b)$ tal que $f'(\alpha) = 0$

En particular, si $f(a) = f(b) = 0 \Rightarrow \exists \alpha \in (a, b)$ tal que $f'(\alpha) = 0$. Mas aun, si $f(a) = f(b) = f(c) = 0 \Rightarrow \exists \alpha \in (a, b) \beta \in (b, c)$ tal que $f'(\alpha) = f'(\beta) = 0$

Teorema del error

Sea f una funcion en $C^{n+1}[a, b]$ y p un polinomio de grado $\leq n$ que interpola a f en $(n + 1)$ puntos distintos $x_0, \ldots, x_n$ en $[a, b]$. Entonces para cada $x \in [a, b]$ $\exists\ \xi = \xi_x \in (a, b)$ tal que

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi) \prod_{i=0}^{n} (x - x_i)$$

Convergencia del polinomio interpolante

A medida que aumenta el grado del polinomio y por lo tanto la cantidad de puntos de interpolacion, el grafico del polinomio tiende a comportarse muy diferente al grafico de la funcion

Diferencias divididas

Teorema

Dados $x_0, \ldots, x_n$ numeros reales distintos, las diferencias divididas satisfacen la siguiente ecuacion:

$$f[x_0, \ldots, x_n] = \frac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0}$$

Tablas

Dados 4 puntos distintos (no necesariamente ordenados) se puede contruir la tabla de diferencias divididas de la siguiente manera:

$$\begin{array}{c|cccc} x_0 & f[x_0] & f[x_0, x_1] & f[x_0, x_1, x_2] & f[x_0, x_1, x_2, x_3] \\ x_1 & f[x_1] & f[x_1, x_2] & f[x_1, x_2, x_3] & \\ x_2 & f[x_2] & f[x_2, x_3] & & \\ x_3 & f[x_3] & & & \end{array}$$

Algoritmo para calcular los coeficientes de la tabla

Para obtener algoritmicamente los coeficientes de la tabla de diferencias divididas se puede pensar la misma como un arreglo matricial donde $c_{ij} = f[x_i, x_{i+1}, \ldots, x_{i+j}]$

```
for j := 1 to n do
  for i := 0 to n-j do
    c_ij ← (c_{i+1,j-1} - c_{i,j-1}/(x_{i+j} - x_i)
  od
od
```

El algoritmo puede ser optimizado almacenando los coeficientes en un vector en vez de una matriz.

Propiedades

Permutacion de los puntos de interpolacion Sean $x_0, \ldots, x_n$ numeros reales distintos y $z_0, \ldots, z_n$ un reordenamiento de $x_0, \ldots, x_n$

$\Rightarrow f[z_0, \ldots, z_n] = f[x_0, \ldots, x_n]$

Error de interpolacion Sea p el polinomio de grado $\leq n$ que interpola a f en los $n+1$ nodos distintos $x_1, \ldots, x_n$. Si t es un numero real distinto de los nodos, entonces:

$$f(t) - p(t) = f[x_0, \ldots, x_n, t] \prod_{j=0}^{n} (t - x_j)$$

Diferencias divididas y derivadas de orden superior Si f es una funcion n veces continuamente diferenciable en $[a, b]$ y $x_0, \ldots, x_n$ son $n+1$ nodos distintos en $[a, b] \Rightarrow \exists$ un punto $\xi \in (a, b)$ tal que

$$f[x_0, \ldots, x_n] = \frac{1}{n!} f^{(n)}(\xi)$$

Interpolacion de Hermite

Tabla de la forma de Hermite

El polinomio interpolante que usa las derivadas en un punto se llama forma de Hermite:

$$\begin{array}{c|cccc} x_0 & f[x_0] & f'(x_0) & f[x_0, x_0, x_1] & f[x_0, x_0, x_1, x_1] \\ x_0 & f[x_0] & f[x_0, x_1] & f[x_0, x_1, x_1] & \\ x_1 & f[x_1] & f'(x_1) & & \\ x_1 & f[x_1] & & & \end{array}$$

El polinomio interpolante basado en la tabla está dado por:

$$p(x) = f[x_0] + f'(x_0)(x - x_0) + f[x_0, x_0, x_1](x - x_0)^2 + f[x_0, x_0, x_1, x_1](x - x_0)^2(x - x_1)$$

Generalizacion Diferencias divididas y derivadas de orden superior

Si f es una funcion n veces continuamente diferenciable en $[a, b]$. Sean $x_0, \ldots, x_n \in [a, b]$ puntos distintos y $z \in (a, b)$. Entonces

$$\lim_{(x_0, x_1, \ldots, x_n) \to (z, z, \ldots, z)} f[x_0, \ldots, x_n] = \frac{f^{(n)}(z)}{n!}$$

Corolario de generalizacion de diferencias divididas y derivadas Si f es n veces continuamente diferenciable en un entorno del punto x_0, entonces:

$$f[x_0, x_0, \ldots, x_0] = \frac{f^{(n)}(x_0)}{n!}$$

Splines

Fenomeno de Runge

Aumentar la cantidad de putnos de interpolacion no mejor la convergencia unforma del polinomio interpolante p_n a la funcion f ### Funcion spline #### Idea Una funcion spline está formada por polinomios definidos en subintervalos, los cuales se unen entre si obedeciendo ciertas condiciones de continuidad

Definicion formal Dados $n + 1$ puntos reales tales que $x_0 < x_1 < \ldots < x_n$, que denominaremos **nodos**, y un entero $k \geq 0$, un **spline de grado k** es una funcion S definida en [x_0, x_n] que satisface:

- S es un polinomio de grado $\leq k$ en cada subintervalo $[x_i, x_{i+1})$, para $i = 0, \ldots, n - 1$
- Las derivadas $S^{(i)}$ son continuas en $[x_0, x_n]$ para $i = 0, \ldots, k - 1$

Splines lineales

Definicion Dados $n + 1$ nodos tales que $x_0 < x_1 < \ldots < x_n$, un **spline lineal** $(k = 1)$ es una funcion S definida en $[x_0, x_n]$ que satisface:

- S es un polinomio de grado ≤ 1 (recta) en cada subintervalo $[x_i, x_{i+1}]$ para $i = 0, \ldots, n - 1$
- La funcion S es continua en $[x_0, x_n]$

Es decir,

$$S(x) = \begin{cases} S_0(x) = a_0 x + b_0, & x \in [x_0, x_1) \\ S_1(x) = a_1 x + b_1, & x \in [x_1, x_2) \\ \vdots & \vdots \\ S_{n-1}(x) = a_{n-1} x + b_{n-1}, & x \in [x_{n-1}, x_n) \end{cases}$$

donde los $2n$ coeficientes a_i, b_i, para $i = 0, \ldots, n-1$ son las incognitas a ser determinadas.

Para eso se deben tener $2n$ condiciones

Error Supongamos que f es 2 veces continuamente diferenciable en $[a, b]$ y $x_k = a + kh, k = 0, \ldots, n,$ con $h = (b-a)/n$

Si S es un spline lineal, en cada intervalo $[x_k, x_{k+1}]$ se tiene un polinomio de grado ≤ 1. Entonces el error de interpolacion para cada $x \in [a, b]$ está dado por:

$$|e(x)| \leq \frac{M}{8} h^2$$

donde $|f''(x)| \leq M \quad \forall x \in [a, b] = [x_0, x_n]$

FALTA SPLINES CUBICOS

Aproximacion de funciones

Aproximacion discreta por Cuadrados minimos

Desviacion absoluta

$$E_1(a_0, a_1) = \sum_{i=1}^{10} |y_i - (a_1 x_i + a_0)|$$

Metodo de cuadrados minimos

Idea El metodo de cuadrados minimos para ajustar a una recta con m datos consiste en determinar a_0 y a_1 tales que minimicen la funcion

$$E(a_0, a_1) = E_2(a_0, a_1) = \sum_{i=1}^{m} [y_i - (a_1 x_i + a_0)]^2$$

con respecto a las variables a_0 y a_1

Condicion para obtener minimo Una condicion necesaria para tener un minimo es que las derivadas parciales de E con respecto a a_0 y a_1 deben ser cero, esto es:

$$\frac{\partial}{\partial a_0} = \frac{\partial}{\partial a_0} \sum_{i=1}^{m} [y_i - (a_1 x_i + a_0)]^2 = 2 * \sum_{i=1}^{m} (y_i - a_1 * x_i - a_0)(-1) = 0$$

$$\frac{\partial}{\partial a_1} = \frac{\partial}{\partial a_1} \sum_{i=1}^{m} [y_i - (a_1 x_i + a_0)]^2 = 2 * \sum_{i=1}^{m} (y_i - a_1 * x_i - a_0)(-x_i) = 0$$

Ecuacion normal Reordenando la ecuacion en la que las derivadas valen 0 se puede obtener el siguiente sistema lineal de dos ecuaciones con las 2 incognitas a_0 y a_1:

$$\begin{cases} a_0 m + a_1 \sum_{i=1}^{m} x_i = \sum_{i=1}^{m} y_i \\ a_0 \sum_{i=1}^{m} x_i + a_1 \sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} x_i y_i \end{cases}$$

Resolver ecuacion normal

$$a_0 = \frac{\left(\sum_{i=1}^{m} x_i^2\right)\left(\sum_{i=1}^{m} y_i\right) - \left(\sum_{i=1}^{m} x_i y_i\right)\left(\sum_{i=1}^{m} x_i\right)}{m\left(\sum_{i=1}^{m} x_i^2\right) - \left(\sum_{i=1}^{m} x_i\right)}$$

$$a_1 = \frac{\left(\sum_{i=1}^{m} x_i y_i\right) - \left(\sum_{i=1}^{m} x_i\right)\left(\sum_{i=1}^{m} y_i\right)}{m\left(\sum_{i=1}^{m} x_i^2\right) - \left(\sum_{i=1}^{m} x_i\right)}$$

Ecuacion normal con integral Sea $f \in C[a, b]$ se desea determinar el mejor polinomio $P_n(x)$ de grado $\leq n$

$$\sum_{k=0}^{n} a_k \int_a^b x^{k+j}\, dx = \int_a^b x^j f(x)\, dx \qquad \text{para} \quad j = 0, \ldots, n$$

Calcular coeficientes de matriz de coeficientes

$$a_{jk} = \int_a^b x^{j+k}\, dx = \frac{b^{j+k+1} - a^{j+k+1}}{j+k+1}$$

Teoremas

Dependencia lineal El conjunto de funciones $\{\phi_0, \ldots, \phi_n\}$ es **linealmente independiente** en el intervalo $[a, b]$, siempre que

$$c_0\phi_0(x) + c_1\phi_1(x) + \cdots + c_n\phi_n(x) = 0 \qquad \text{para cualquier } x \in [a, b]$$

Se tiene que $c_0 = c_1 = \cdots = c_n = 0$

En caso contrario se dice que ese conjunto de funciones es **linealmente dependiente**

Polinomio e independencia lineal Si $\phi_j(x)$ es un polinomio en x de grado igual a j para $j = 0, \ldots, n$, entonces $\{\phi_0, \ldots, \phi_n\}$ es un conjunto linealmente independiente para cualquier intervalo $[a, b]$

Polinomio y combinacion lineal Si $\{\phi_0, \ldots, \phi_n\}$ es un conjunto de polinomios linealmente independiente para cualquier intervalo $[a, b]$ en el espacio de polinomios de grado $\leq n$, entonces todo polinomio de grado $\leq n$ puede escribirse, de manera unica, como combinacion lineal de $\{\phi_0, \ldots, \phi_n\}$