Resumen Primer Parcial - Análisis Numérico

Lautaro Bachmann

Contents

Introduccion

Preliminares matematicos

Valor intermedio para funciones continuas

Sea f continua en $[a, b]$. Sea d entre $f(a)$ y $f(b)$

$\Rightarrow \exists c \in [a, b]$ tal que $f(c) = d$

Valor medio

Sea f continua en $[a, b]$ y derivable en (a, b).

$\Rightarrow$ 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 $f(x) = f(c) + f'(\epsilon)(x - c)$

Es decir, la derivada en c es igual a la pendiente de la recta que une los puntos $(x, f(x))$ y $(c, f(c)$

Taylor

Definicion 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:

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

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

Taylor con resto integral Si $f \in C^{(n)}[a, b]$ y existe $f^{(n+1)}(a, b)$

$\Rightarrow$ para todo par $x, c \in [a, b]$ se tiene que:

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

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 $n \geq r$

Esta notacion tambien se puede usar para comparar funciones.

$f(x) = O((g(x))$ cuando $x \to \infty \Leftrightarrow |f(x)| \leq C|g(x)|$

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 $\lim_{n\to\infty} \left(\frac{x_n}{\alpha_n}\right) = 0$

Esta notacion tambien sirve para comparar funciones:

$f(x) = o(g(x))$ cuando $x \to \infty \Leftrightarrow \lim_{x\to\infty} \left(\frac{f(x)}{g(x)}\right) = 0$

Comparar funciones cuando tienden a su punto de convergencia

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 $\lim_{x \to x_*} \left(\frac{f(x)}{g(x)} \right) = 0$

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 requiren 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

Errores absolutos y relativos

Error absoluto

Sean r el valor exacto y $\bar{r}$ una aproximacion de r

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

Error relativo

Sean r el valor exacto y $\bar{r}$ una aproximacion de r

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

Error relativo porcentual

$100 * \delta r$

Redondeo y truncado

Redondeo

Definicion 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 Sean r el valor exacto, $\bar{r}$ una aproximacion de r y n el numero de digitos

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

Ejemplo: Redondear $r = 0.11$, con $n = 1$ $\quad r = 0.11 \Rightarrow \bar{r} = 0.1 \Rightarrow |r - \bar{r}| = 0.01$

$\Rightarrow 0.01 \leq 0.05 \Rightarrow 0.01 \leq 5 \cdot 10^{-2}$

$\Rightarrow 0.01 \leq \frac{1}{2} 10^{-1}$

Truncado

Definicion 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: Truncar $r = 0.11$, con $n = 1$ $\quad r = 0.11 \Rightarrow \bar{r} = 0.1 \Rightarrow |r - \bar{r}| = 0.01$

$\Rightarrow 0.01 \leq 0.1$

$\Rightarrow 0.01 \leq 10^{-1}$

Digitos significativos

Definicion

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

$\delta r \leq 5 \cdot 10^{-m}$

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

Erorres en las operaciones

Suma

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

$|y - \bar{y}| \leq |x_1 - \bar{x_1}| + |x_2 - \bar{x_2}|$

Es decir,

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

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

Resta

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

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

Es decir,

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

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

General 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

Definicion

Cuando se restan dos numeros cercanos se genera un error bastante grande.

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

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:

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

con $d_{-i} \in \{0, \ldots, \beta - 1\}$ para $i = 1, \ldots t$, 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, $\dfrac{1}{\beta} \leq |m| < 1$

Observaciones

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

Representacion del 0 El cero no puede representarse en este sistema de numeros normalizados

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

Errores de redondeo

Representacion como numero flotante Sea $x = m\beta^e, \qquad \dfrac{1}{\beta} \leq |m| < 1,$

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 Error absoluto de representacion en x es:

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

Error relativo 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 $|m| \geq \dfrac{1}{\beta} \quad \Rightarrow \quad \dfrac{1}{|m|} \leq \beta$

Error relativo acotacion 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

Solucion de ecuaciones no lineales

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

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**
  $e \leftarrow \dfrac{e}{2}$
  $c \leftarrow a + e$
  $w \leftarrow f(c)$
  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 $c_n = \dfrac{1}{2}(a_n + b_n)$ y $r = \lim_{n\to\infty} c_n$,

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

Relacion elementos finales e iniciales

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

Metodo de Newton

Idea

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

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

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
$i \leftarrow 1$
while $i \leq M$ **do**
  $p \leftarrow g(p_0)$
  output i,p
  if $|p - p_0| < \delta$ **then** STOP **fi**
  $i \leftarrow i + 1$
  $p_0 \leftarrow p$
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:

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

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

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

Caracteristicas

Existencia y unicidad

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

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 $x_0, \dots, x_n$ en $[a, b]$. Entonces para cada $x \in [a, b]$ $\exists\ \xi = \xi_x \in (a, b)$ tal que

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