MÈTODES NUMÈRICS I

Grau de Matemàtiques. Curs 2018-19. Semestre de primavera

Pràctica 3: Interpolació polinomial

Exercici 1 Interpolació de Lagrange

a) Escriviu una funció amb prototipus

```
void dif_div(int n, double *x, double *y, double *difer);
```

que, donats els vectors `x` i `y`, que contenen $\{x_0,\dots,x_n\}$ i $\{f(x_0),\dots,f(x_n)\}$, respectivament, omple el vector `difer` amb les diferències dividides

$$f[x_0], f[x_0,x_1],\dots, f[x_0,\dots,x_n],$$

en aquest ordre. Guardeu-la en un fitxer de nom `difDiv.c`.

Feu una funció `main` per comprovar que aquesta funció funciona. Executeu el programa per a diferents taules de valors.

b) Escriviu una funció amb prototipus

```
double aval(int n, double *x, double *difer, double z);
```

que, donats el vector `x`, que conté $\{x_0,\dots,x_n\}$, i el vector `difer`, que conté les diferències dividides $\{f[x_0], f[x_0,x_1],\dots, f[x_0,\dots,x_n]\}$, avaluï el polinomi interpolador en el punt `z`, usant la regla de Horner.

Recordeu que en el mètode de Newton el polinomi interpolador ve donat per

$$p_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots + a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})$$

amb $a_j = f[x_0,x_1,\dots,x_j],\ j = 0,\dots,n.$

Verifiqueu aquesta funció per a diferents valors de `z`. Guardeu-la en un fitxer de nom `hornerDif.c`.

c) Feu una funció que calculi l'error màxim entre el polinomi interpolador en m punts i la funció interpolada f en un interval donat $[a,b]$. Caldrà fer una xarxa de `num` punts (almenys 200) a l'interval $[a,b]$. La funció tindrà el prototipus

```
double testLagrange(int m, double a, double b, double num)
```

i retornarà l'error màxim comés en els punts de la xarxa.

Aplicacions

1 Volem interpolar la funció $g(x) = (1+x)^{1/2}$, $x \in [-0.9, 0.5]$ en m abscisses equidistants per a $m = 5, 10, 15, 20$. Compareu l'error obtingut per `testLagrange` amb la fita teòrica de l'error.

2 Fenomen de Runge. Interpolem la funció

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

per un polinomi de grau m a l'interval $[-b,b]$ en abscisses equidistants. L'anomenem p_m^b. Feu-lo per a $m = 11, 21, 31$ i per a $b = 5 - 10^{-1}\cdot j,\ j = 1,\dots,9$. Calculeu l'error en aproximar f per p_m^b a l'interval $[-b,b]$ fent servir la funció `testLagrange`. Què observeu?

Exercici 2 Interpolació d'Hermite

a) Programeu una funció amb prototipus

```
void dif_herm(int n, double *x, double *fx, double *derfx,
              double *diferH);
```

que, donats els vectors `x`, `fx` i `derfx`, que contenen els valors $\{x_0,\dots,x_n\}$, $\{f(x_0),\dots,f(x_n)\}$ i $\{f'(x_0),\dots,f'(x_n)\}$ respectivament, omple el vector `diferH` amb les diferències dividides

$$f[x_0], f[x_0,x_0], f[x_0,x_0,x_1],\dots, f[x_0,x_0,\dots,x_n,x_n],$$

en aquest ordre. Guardeu-la en un fitxer de nom `difDivHermite.c`.

Verifiqueu aquesta funció per a diferents taules de valors.

b) Per avaluar el polinomi interpolador d'Hermite en el punt `z`, usant la regla de Horner, useu la funció `aval` fent una crida adequada.

Feu una funció `main` per comprovar que aquesta funció funciona. Executeu el programa per a diferents taules de valors.

Aplicació

Volem aproximar e^x a l'interval $[0,1]$ per un polinomi d'interpolació de Hermite en $n+1$ abscisses equiespaiades. Determineu un valor de n que asseguri que l'error d'interpolació serà inferior a 10^{-12}.

Usant la funció `exp` de l'ordinador per omplir la taula inicial, calculeu el polinomi d'interpolació d'Hermite en aquestes abscisses. Feu una malla de pas `h` a l'interval indicat i comproveu que l'error obtingut és menor de 10^{-12}.

Exercici 3 La primera derivada d'una funció $f(x)$ en un punt x_0 es pot aproximar, si h és prou petit, per les expressions següents:

$$F_1(x_0,h) = \frac{f(x_0+h)-f(x_0)}{h} \quad \text{i} \quad F_2(x_0,h) = \frac{f(x_0+h)-f(x_0-h)}{2h}.$$

a) Programeu les funcions de prototipus

```
double primDeriv(double x0, double h);
double primDerivCentr(double x0, double h);
```

que, donats x_0 i h, retornin el quocient incremental $F_1(x_0,h)$ i $F_2(x_0,h)$, respectivament. Guardeu-les en un fitxer de nom `primeraDerivada.c`.

b) Programeu una funció `main` que llegeixi x_0, h_0, r i n, avaluï $F_1(x_0,h_i)$ i $F_2(x_0,h_i)$ per a diferents passos $h_i = \dfrac{h_{i-1}}{r}$, $i = 1,\dots,n$, i escrigui en un fitxer la informació següent (en punt flotant amb notació exponencial i controlant el nombre de dígits de la mantissa):

$$h_i \qquad |f'(x_0)-F_1(x_0,h_i)| \qquad |f'(x_0)-F_2(x_0,h_i)|$$

Guardeu-la en un fitxer de nom `mainDerivada.c`.

Per calcular el valor de f i de f' caldrà programar dues funcions de prototipus `double f(double);` i `double df(double);` que retornin el valor de $f(x)$ i $f'(x)$, respectivament.

c) Feu una gràfica dels errors, en valor absolut, en funció del pas h usant una escala logarítmica. Deduïu-ne el comportament de l'error en l'aproximació amb cadascuna de les fórmules.

Aplicacions

Calculeu aproximacions a les derivades de les funcions en els casos següents:

a) $f(x) = \sin^2(x),\ x_0 = 1.8,\ h_0 = 1;$

b) $f(x) = \ln x$, i) $x_0 = 2,\ h_0 = 1$; ii) $x_0 = 0.3,\ h_0 = 0.2;$

c) $f(x) = e^{x^3},\ x_0 = 1,\ h_0 = 0.2.$

Proveu per diferents valors d'r per calcular una aproximació del pas òptim.

Exercici 4 Volem calcular, per a diferents valors de α,

$$\int_0^1 \cos\,(\alpha x^2)\;dx\,.$$

a) Feu una estimació de la integral per a valors de α petits usant pocs termes de la sèrie de Taylor de $\cos\,(\alpha x^2)$ i integrant terme a terme.

b) Per a $\alpha = 1$, estudieu com depèn l'error de discretització del pas en les fórmules dels trapezis i en la de Simpson compost. Feu una taula amb els resultats per a diferents passos i expliqueu perquè els vostres resultats mostren aquesta dependència.

c) Dibuixeu l'integrand, $\cos(\alpha x^2)$, per $\alpha = 0.1, 1, 10$. Per a quin d'aquests integrands penseu que serà més difícil de calcular la integral?

d) Per aproximar $\int_a^b f(x)dx$ usem ara la fórmula dels trapezis. Prenent el pas $h = \frac{b-a}{n}$ i les abscisses $x_i = a + ih\ (i = 0, \ldots, n)$, llavors la fórmula dels trapezis és

$$T(h) = h\left[\frac{f(x_0)}{2} + f(x_1) + \ldots + f(x_{n-1}) + \frac{f(x_n)}{2}\right]$$

Programeu una funció `double trapezis(double a, double b, int n);` que, donats un interval $[a,b]$, i un enter n retorni una aproximació de la integral de $f(x)$ a l'interval $[a,b]$ usant la fórmula dels trapezis, on n és el nombre de subintervals en què dividim l'interval $[a,b]$.

Exercici 5 Apliqueu l'extrapolació de Richardson a alguns dels exemples dels exercicis 3 i 4. Feu una taula d'aproximacions i pareu quan dues aproximacions siguin prou properes. Podeu fer els càlculs a mà o programar una funció.