ICC Pràctica 2: Interpolació polinomial

Objectiu: L'objectiu d'aquesta pràctica és implementar en C el mètode de les diferències dividides de Newton per a obtenir un polinomi interpolador i veure'n alguns exemples.

Estructurarem l'enunciat de la pràctica en diferents parts per tal de facilitar-ne la comprensió i la implementació de les funcions. Les diferents parts s'aniran explicant en les successives classes de laboratori d'ordinadors i l'enunciat s'anirà actualitzant progressivament.

Important: No es corregiran les entregues que no segueixin les indicacions indicades al llarg de l'enunciat i/o en les classes de laboratori d'ordinadors.

Organització dels fitxers:

- En començar la pràctica creeu un directori `Cognom1Cognom2Nom_prac2` on anireu creant els diferents arxius en C de la pràctica (amb el nom que s'indica a l'enunciat).
- Les funcions principals estaran en fitxers `main_XXX.c`. La resta de funcions necessàries per compilar i executar els codis relatius a la resolució de sistemes lineals en el fitxer `funs_interp.c`.

Instruccions per entregar: A la corresponent tasca del Campus Virtual s'entregarà un únic fitxer `Cognom1Cognom2Nom_prac2.tgz` que contindrà *exclusivament*:

- Els arxius .c que s'indiquen al llarg de l'enunciat amb el codi C de les funcions programades per a la realització de la pràctica.
- El fitxer `funs_interp.h` amb les capçaleres de les funcions.
- Un fitxer `.pdf` amb una petita explicació del que s'ha fet, els resultats obtinguts, comprovacions fetes, detalls d'implementacions, respostes dels enunciats, etc. També cal afegir comentaris relatius a les qüestions **Q** que apareixen en l'enunciat.

Assegureu-vos que no hi ha cap altre arxiu en el directori: elimineu objectes, executables, ocults, etc.

Per crear l'arxiu `.tgz` executeu des del directori pare la comanda

```
tar -czvf Cognom1Cognom2Nom_prac2.tgz ./Cognom1Cognom2Nom_prac2/
```

Data (límit) d'entrega: Divendres 10 de Desembre de 2021 a les 23:55h.

---

1. Implementeu una funció que avaluï un polinomi usant l'algorisme de Horner amb capçalera

```
double horner(double z, double *x, double *c, int n )
```

  - Rebrà com a paràmetres: el valor on volem avaluar el polinomi `z`, el vector `x`$= (x_0, x_1, \ldots, x_n)$ d'abcisses, i el vector `c`$= (c_0, c_1, \ldots, c_n)$ de coeficients.
  - La funció retornarà el valor

$$p(z) = \sum_{i=0}^{n} c_i \left(\prod_{j=0}^{i-1} (z - x_j) \right) .$$

  Recordem que el mètode de Horner calcula $p(z)$ de la següent forma:

$$p = c_n \ , \qquad \forall i = n-1, n-2, \ldots, 1, 0 \quad p \leftarrow p * (z - x_i) + c_i \ .$$

2. Implementeu una funció amb capçalera

```
int difdiv(double *x, double *f, int n)
```

per a calcular les diferències dividides de Newton.

  - Rebrà com a paràmetres: les abcisses `x`$= (x_0, x_1, \ldots, x_n)$ i els valors de la funció a interpolar en les abcisses `f`$= (f_0, f_1, \ldots, f_n)$.
  - A la sortida, el vector `f` contindrà les diferències dividides associades a la taula de valors (x_i, f_i) , $i = 0, 1, \ldots, n$.
  - Si el procés s'ha pogut fer sense cap entrebanc, la funció retorna el valor 0. En canvi, si algun dels denominadors que surten en el procés té valor absolut menor que 10^{-12} llavors el procés no continua i la funció retorna el valor -1.

Les diferències dividides s'obtenen de forma recursiva com

$$f[i] = (f[i] - f[i-1])/(x[i] - x[i-k]) \quad \forall i = n, n-1, \ldots, k \ ; \ \forall k = 1, 2, \ldots, n \ .$$

Feu servir la forma recursiva anterior en la implementació.

Q: Com es relaciona la forma recursiva anterior amb l'esquema triangular del càlcul de les diferències dividides de Newton (p.11 slides de teoria)?

3. Implementeu una funció principal `main_taula.c` que llegeixi les abcisses i les ordenades d'un fitxer, llegeixi el grau del polinomi n, llegeixi els extrems d'un interval $[a, b]$, i escrigui en un fitxer el resultat d'avaluar el polinomi interpolador en una xarxa de 1000 punts equidistants en $[a, b]$.

   Useu la funció anterior per representar el polinomi interpolador $p_3(x)$ corresponent a la taula

x_k	1	2.7	3.2	4.8
$f(x_k)$	14.2	17.8	22.0	38.3

   Representeu $p_3(x)$ i els punts d'interpolació usant gnuplot.

   Indicacions: Les dades corresponents es llegiran del fitxer `taula.in`, que contidrà dues columnes amb x_k i $f(x_k)$ respectivament. Nom del fitxer de sortida: `p3taula.out`.

   Podeu representar la "corba" corresponent al polinomi interpolador i els punts de la taula a la vegada fent

   ```
   plot 'p3taula.out' w d, 'taula.in' w p pt 7 lc 2 ps 1.5
   ```

   Si guardeu la comanda de gnuplot anterior en el fitxer `taula.gnu` llavors podeu recuperar el dibuix escrivint `load 'taula.gnu'` des del `gnuplot`.

4. L'objectiu d'aquest exercici és estudiar l'error en interpolar per un polinomi de grau n una determinada funció f en un interval $[a, b]$.

   Per a fer-ho, implementeu una funció principal `main_errinterp.c` que llegeixi el grau del polinomi d'interpolació n i els extrems de l'interval $[a, b]$ on s'interpolarà la funció usant nodes equidistants i el nom del fitxer de sortida. Finalment, considerarà una xarxa de 1000 punts equidistants z_j en $[a, b]$ i escriurà en el fitxer de sortida amb estructura:

$$z_j \quad f(z_j) \quad p_n(z_j)$$

   L'última línia del fitxer de sortida començarà amb el caràcter #, que indica que és un comentari de `gnuplot`, i en la mateixa línia s'escriurà el màxim de $|f(z_j) - p_n(z_j)|$ en la xarxa de punts considerada.

   Useu la funció anterior en els següents casos (la funció principal deixarà triar quina funció es fa servir: `ln` o `runge`):

   (4.1) Considereu la funció logaritme neperià[1] $f(x) = \ln(x)$ i abscisses equiespaides en $[a, b] = [0.4, 0.8]$. Per avaluar $f(x)$, implementeu una funció amb capçalera

```
double fun_log(double z)
```

   Representeu els polinomis $p_n(x)$, $1 \leq n \leq 5$ i les gràfiques de l'error comés.
   Feu 2 o 3 representacions de l'error en rangs de y adequats[2].
   Noms del fitxers de sortida: `log1.out`, ... , `log5.out`.

   Q: És compatible l'error d'interpolació observat amb la fita de l'error en nodes equiespaiats (p.16 slides teoria)? Calculeu la fita i compareu.

   (4.2) Considereu la funció $f(x) = 1/(1 + 25x^2)$ i abcisses equiespaiades a l'interval $[a, b] = [-1, -1]$. Per avaluar $f(x)$, implementeu una funció amb capçalera

```
double fun_runge(double z)
```

   Representeu els polinomis $p_n(x)$, $1 \leq n \leq 10$ i les gràfiques de l'error comés.
   Noms del fitxers de sortida: `runge1.out`, ..., `runge10.out` [3]

   Q: Llegiu les pàgines 18 i 19 de les slides de teoria i comenteu els resultats que obteniu en (4.2). És bona idea considerar molts nodes d'interpolació en un interval per tenir una millor aproximació?

[1] La funcio `log()` de la llibreria `math.h` retorna el logaritme neperià.

[2] Pot ser convenient guardar les comandes de gnuplot per obtenir les representacions en un fitxer `log.gnu`. Useu `set yrange [a:b]`, amb valors de `a` i `b` adequats, per fixar el rang de y. La comanda `set auto` esborra la definició de rangs que s'hagi fet prèviament. Useu `pause -1` entre diferents `plot`.

[3] Pot ser convenient guardar les comandes de gnuplot en un fitxer `runge.gnu`.