Pràctica 2: Zeros de funcions

Les instruccions precises de què (i quan) cal entregar com a resultat d'aquesta pràctica, les expliquem al final d'aquest document.

El nostre objectiu és trobar zeros de sistemes d'equacions no lineals $f(x) = 0$ (observeu que aquesta pràctica complementa la primera, on el que fèiem era resoldre equacions lineals: $Ax - b = 0$).

Farem servir el mètode de Newton. És iteratiu, i, en cada iteració se substitueix el problema de resoldre equacions no lineals en el problema de resoldre equacions lineals. Conseqüentment la primera pràctica ens serà molt útil, especialment a la segona part.

Dividim la pràctica en **DUES** parts: el cas d'una variable i el cas de diverses variables.

Primera Part: Cas d'una variable

Suposem que $f : \mathbb{R} \to \mathbb{R}$ és una funció derivable.

a) Escriviu una funció que implementi el mètode de Newton per trobar una solució de $f(x) = 0$ a partir d'una aproximació inicial donada (que suposarem propera al zero que estem buscant). La funció tindrà com a capçalera

```
int Newton(double x, double *sol, double tol, int iter)
```

Així, a la funció se li passen com a paràmetres l'aproximació inicial `x`, la precisió demanada `tol` i el nombre màxim d'iterats permesos `iter`. El *mètode de Newton* és

$$x_0 = x, \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \geq 0$$

Les condicions de retorn de la funció són: si $|x_{n+1} - x_n| <$`tol` o $|f(x_n)| <$`tol`, l'últim iterat s'assignarà a la variable `*sol` i retorna 0; en canvi, si no es compleix alguna de les condicions abans de `iter` vegades o si $|f'(x_n)| <$`tol` retorna 1, indicant que no s'ha trobat una aproximació prou bona del zero.

b) També cal implementar les funcions que avaluen tant f com f' en un punt. Les funcions tindran, respectivament, les capçaleres

```
double Fun(double x)      double dFun(double x)
```

Aquestes contindran les diverses expressions dels exemples i caldrà comentar i descomentar-les adeqüadament per a treballar amb cadascun dels casos.

Exemples (cal que penseu una aproximació inicial x_0 en cada cas):

$$(a)\ f(x) = x^2 + \sin(x) - \pi, \qquad (b)\ f(x) = 1 - \ln(x)$$
$$(c)\ f(x) = \sqrt{x} - e^{-x}, \qquad (d)\ f(x) = \frac{e^x - e^{-x}}{2} - \sin(x)$$

c) La funció `main` s'encarregarà de llegir la tolerància, el nombre màxim d'iteracions i l'aproximació inicial, cridarà `Newton` i escriurà el resultat corresponent.

Caldrà elaborar un fitxer `newton1.c` amb les funcions `main`, `Newton`, `Fun` i `dFun`.

Aplicació: Trobar conques d'atracció de zeros de polinomis

Sigui $P(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ ($a_n \neq 0$) un polinomi de grau n. Se sap que totes les seves arrels reals (n'hi ha una quantitat entre 0 i n) estan a l'interval $I = [-M, +M]$ amb $M = 1 + \displaystyle\sum_{j=0}^{n-1} \left|\frac{a_j}{a_n}\right|$.

Donat un zero α de P, la seva conca d'atracció C_α és el conjunt de valors $x_0 \in \mathbb{R}$ tals que, quan apliquem el mètode de Newton amb aproximació inicial x_0, convergim a α. En general, una conca d'atracció és una reunió d'intervals.

L'objectiu és trobar els zeros de P i les seves conques d'atracció respectives. El procediment serà el següent: Donat un pas h (per exemple $10^{-k}, k = 3, 4, 5, 6$), anem aplicant el mètode de Newton amb aproximacions inicials a l'interval I, separades una distància h.

Exemple: $P(x) = x^2 - 1$, $M = 2$. S'obté que si $x_0 \in [-2, -h]$ es convergeix a l'arrel -1, si $x_0 \in [h, 2]$ convergeix a l'arrel +1, i per $x_0 = 0$, no hi ha convergència.

Elaboreu un fitxer `conca.c` a partir de l'anterior, amb les funcions `Fun` i `dFun` corresponents a cada exemple, i la funció `main` llegirà també la constant M, calculada prèviament a mà, i el pas h. Partint de $x_0 = -M$ i amb pas h anirà cridant `Newton` i escrivint els resultats.

Apliqueu-ho als següents polinomis:

$$(a)\ P(x) = x^2 - 1,\ M = 2 \qquad (b)\ P(x) = x^3 - x,\ M = 2$$
$$(c)\ P(x) = 3x^3 - x + 1,\ M = 5/3 \qquad (d)\ P(x) = x^4 + 1,\ M = 2$$
$$(e)\ P(x) = \prod_{i=1}^{6}(x - \frac{10i}{i+1}),\ M =? \qquad (f)\ P(x) = \prod_{i=1}^{6}(x - \frac{i+1}{10i}),\ M =?$$

Segona Part: Diverses variables

En la segona part de la pràctica volem resoldre un sistema d'equacions no lineals fent servir el mètode de Newton en diverses variables. Només considererem el cas de dimensió tres:

$$\begin{aligned} f_1(x_1, x_2, x_3) &= 0 \\ f_2(x_1, x_2, x_3) &= 0 \\ f_3(x_1, x_2, x_3) &= 0 \end{aligned} \tag{1}$$

Si escrivim $f = (f_1, f_2, f_3)$ i $x = (x_1, x_2, x_3)$ volem resoldre $f(x) = 0$. Recordem que el mètode de Newton per funcions de varies variables s'escriu com:

$$x^{(n+1)} = x^{(n)} - \Delta x^{(n)}$$

on $\Delta x^{(n)}$ és la solució del sistema lineal d'equacions

$$Df(x^{(n)})\Delta x^{(n)} = f(x^{(n)}). \tag{2}$$

Aquí $Df(x^{(n)})$ és la matriu de les derivades parcials de la funció f (en aquest casi, 3×3).

Per tant per anar de $x^{(n)}$ a $x^{(n+1)}$ el que cal és resoldre el sistema lineal (2). I un cop tens $\Delta x^{(n)}$ simplement substitueixes $x^{(n)}$ per $x^{(n+1)} = x^{(n)} - \Delta x^{(n)}$. Però resoldre sistemes lineals és el que vàrem programar en la pràctica 1 fent servir la funció `gausspivot` (i, per tant, també `resoltrisup`). Assegureu-vos que la vostra funciona correctament.

Les capçaleres de les funcions de la primera part han de ser ara:

```
void Fun(double *x, double *f)      void dFun(double *x, double **df)
int Newton(double *x, double *sol, double tol, int iter)
```

O sigui, les diferències són: a cada iteració de Newton cal resoldre un sistema lineal usant `gausspivot`; els valors absoluts en dimensió 1 s'han de substituir per la norma vectorial en dimensió 3; la condició sobre la derivada no cal avaluar-la ja que equival a no poder resoldre el sistema lineal.

Considerarem només aproximacions inicials dins d'un cert cub donat
$Q = \{(x, y, z) \in \mathbb{R}^3 \mid x \in [a_1, b_1],\ y \in [a_2, b_2],\ z \in [a_3, b_3]\}$.
Feu un `main` que llegeixi la tolerància, el nombre màxim d'iteracions i el cub; generi 100 condicions inicials a l'atzar (`http://www.gnu.org/software/libc/manual/html_node/ISO-Random.html`) dins del cub i, per a cada una, iteri el mètode de Newton i escrigui la solució on convergeix (si ho fa).

Apliqueu-ho als sistemes següents:

$$\left\{\begin{array}{l} x + y + z = 1 \\ \quad y + z = 0 \\ x^2 + 0.75y = 0 \end{array}\right. ,\ Q = [-1, 1]^3 \qquad \left\{\begin{array}{c} x^2 + y^2 + z^2 = 1 \\ \frac{1}{4}(x - y)^2 + (x + y)^2 + z^2 = 1 \\ (x - y)^2 + (x + y)^2 + \frac{1}{4}z^2 = 1 \end{array}\right. ,\ Q = [-1, 1]^3$$

Totes les funcions estaran en un fitxer `newton3.c`.

Instruccions per a l'entrega

Abans de començar a fer la pràctica heu de crear un subdirectori anomenat:

Grup-Cognom1Cognom2Nom-X

on

- Grup: és el vostre grup de pràctiques en majúscules (pot ser A, B, C o D).
- Cognom1Cognom2Nom: és el vostre primer cognom, segon cognom i nom.
- X: identifica el número de la pràctica (1, 2, 3, etc).

Exemple: `A-LopezPerezMaria-2` correspon a una alumna del grup A que comença a fer la pràctica 2.

Aquest directori contindrà els arxius .c corresponents a les diverses parts.

Es crearà un arxiu comprimit del directori amb la comanda
`tar -czvf A-LopezPerezMaria-2.tgz A-LopezPerezMaria-2`
executada des del directori pare.

Entregar la pràctica vol dir el següent:

(1) S'entregarà el fitxer comprimit (.tgz) al Campus Virtual (moodle). La data límit serà les 23 hores del dia 16 de desembre.

Tots els programes lliurats hauran de començar amb el comentari següent, on cadascun dels alumnes inclourà les seves dades personals

/* COGNOM1: COGNOM2: NOM: DNI: */

Tots els programes lliurats hauran de compilar amb les opcions: `-ansi`, `-pedantic`, `-O` i `-Wall`.

(2) El dimarts 18 de desembre a l'hora de laboratori es proposarà un problema (el que hem anomenat **prova** en el pla docent) que s'haurà de resoldre usant exclusivament les funcions (programes) que heu el·laborat durant la pràctica. S'entregarà un fitxer .c al Campus Virtual.