Pràctica 1: Àlgebra Lineal

Al final del document s'explica amb detall què (i quan) cal entregar en aquesta pràctica.

Fixades unes dimensions $m > n \geq 2$ volem trobar "solucions" de sistemes sobredeterminats (més equacions que incògnites).

Primera Part: Sistemes triangulars

Implementeu una funció per resoldre el sistema d'equacions $Lx = b$ quan L és triangular inferior (amb zeros sobre la diagonal principal) amb 1 a la diagonal. La funció tindrà com a capçalera

```
void resTinf (int n, double **L, double *x, double *b);
```

La funció rebrà com a paràmetres la dimensió `n`, la matriu L i el vector b, respectivament.

Per tal de resoldre el sistema, la funció `resTinf` suposarà que la matriu L és triangular inferior, és a dir, tal que $l_{ij} = 0$ per $i < j < n$, $i = 0, \ldots, n-1$ i amb $l_{ii} = 1$ per a $i = 0, \ldots, n-1$. La solució del sistema es guardarà en el vector `x`. Cal usar les fórmules de substitució endavant:

$$x_0 = b_0, \qquad x_i = b_i - \sum_{k=0}^{i-1} l_{ik}\, x_k \quad i = 1, 2, \ldots, n-1$$

Dissenyeu un programa principal (la funció `main`) que llegeixi una matriu $L = (l_{ij})_{0 \leq i,j < n}$ (comprovant que té 1 a la diagonal), llegeixi un vector $b = (b_i)_{0 \leq i < n}$ i cridi la funció `resTinf`. Una vegada s'hagi calculat la solució x del sistema $Lx = b$, el programa principal escriurà el vector solució x i el valor residual $\|Lx - b\|_2$ com a indicador de la bondat de la solució x calculada.

Ara voldrem resoldre sistemes triangulars superiors de la forma $Ux = b$ amb U triangular superior amb 1 a la diagonal. Implementeu la funció

```
void resTsup (int n, double **U, double *x, double *b);
```

amb els paràmetres similars a `resTinf`. Cal usar les fòrmules de sustitució enrere

$$x_{n-1} = b_{n-1}, \qquad x_i = b_i - \sum_{k=i+1}^{n-1} u_{ik}\, x_k \quad i = n-2, \ldots, 1, 0$$

La funció `main` transposarà la matriu triangular inferior llegida i resoldrà el sistema triangular superior amb el mateix terme independent. Com a aplicació resoleu els sistemes següents.

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ 4 & 3 & 2 & 1 \end{pmatrix}, b = \begin{pmatrix} -4 \\ -5 \\ -6 \\ -2 \end{pmatrix}, L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2.0981 & 1 & 0 & 0 \\ 9.9871 & 2.2222 & 1 & 0 \\ 1.1 & 0.3333 & 20.121 & 1 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

Per calcular el producte matriu per vector $y = Ax$ $y \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$ useu la funció

```
void prodMatVec (int m, int n, double **A, double *x, double *y);
```

Segona Part: Descomposició LDL^t

Donada una matriu $A \in \mathbb{R}^{n\times n}$ simètrica ($a_{ij} = a_{ji}$) en determinades condicions pot descompondre's en la forma $A = LDL^t$, amb L triangular inferior amb 1's a la diagonal i D matriu diagonal. Una vegada trobada la descomposició, per resoldre $Ax = (LDL^t)x = b$ resoldrem, en aquest ordre, $Lz = b$, $Dy = z$ i $L^t x = y$; els primer i tercer sistemes són triangulars i el segon, diagonal (de solució trivial).

Les fórmules per trobar L i D són, per a $k = 0, \cdots, n-1$:

$$d_{kk} = a_{kk} - \sum_{j=0}^{k-1} l_{kj}^2 d_{jj}, \quad l_{kk} = 1$$

$$l_{ik} = \frac{1}{d_{kk}} \left(a_{ik} - \sum_{j=0}^{k-1} l_{ij}\, l_{kj}\, d_{jj} \right) \quad i = k+1, \cdots, n-1$$

Implementeu una funció que faci aquest algorisme. La funció tindrà com a capçalera

```
int ldlt(int n, double **A, double tol)
```

i descompondrà la matriu $A = LDL^t$; en entrar A conté la matriu del sistema (que suposarem simètrica); en sortir A contindrà la desomposició LDL^t, això és: la part triangular inferior d'A contindrà L, la diagonal contindrà D i la part triangular superior L^t. La funció retornarà 0 si s'ha pogut descomposar i 1 si no ha estat possible (quan $|d_{kk}| < \epsilon$, ϵ una tolerància donada)

Notes:

a) Observeu que no es guarden els 1 de les diagonals de les matrius L i L^t.

b) Observeu que, un cop descompost el sistema, la funció `main` haurà de resoldre el sistema fent càlculs i cridant les funcions escaients.

c) `A` no conté la matriu original a la sortida de la funció.

Dissenyeu un programa principal (la funció `main`) que llegeixi una matriu $A = (a_{ij})_{0\le i,j<n}$, comprovant la simetria, llegeixi un vector $b = (b_i)_{0\le i<n}$ i cridi la funció `ldlt` i resolgui el sistema lineal $Ax = b$, el programa principal escriurà el vector solució x i el valor residual $\|Ax - b\|_2$ com a indicador de la bondat de la solució x obtinguda.

Per testejar la funció resoleu els sistemes

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}, b = \begin{pmatrix} 2 \\ 5 \\ 8 \end{pmatrix} \quad , A = \begin{pmatrix} 1.1 & 2.3 & 4.1 & 0.3 \\ 2.3 & -4.2 & 3.2 & -1 \\ 4.1 & 3.2 & 8.27 & 0 \\ 0.3 & -1 & 0 & 1.23 \end{pmatrix}, b = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

Tercera (i última) Part: Equacions normals

Quan tenim plantejat un sistema $Ax = b$ amb més equacions que incògnites, en general no tindrà solució. Aleshores es busca x^* tal que el valor $\|Ax-b\|_2$ sigui el més petit possible (si existeix solució el valor teòric és 0). És fàcil veure que aquesta x^* és solució del sistema $n\times n$ $A^tAx = A^tb$, anomenat **sistema d'equacions normals** amb la matriu del sistema A^tA simètrica.

Dissenyeu un programa principal (la funció `main`) que llegeixi la matriu $A = (a_{ij})$ $0 \le i < m$, $0 \le j < n$ (amb $m > n$) i el vector $b = (b_i)$ $0 \le i < m$, calculi la matriu A^tA i el vector A^tb usant

```
void prodMatMat (int m, int n, int p, double **A, double **B, double **C);
```

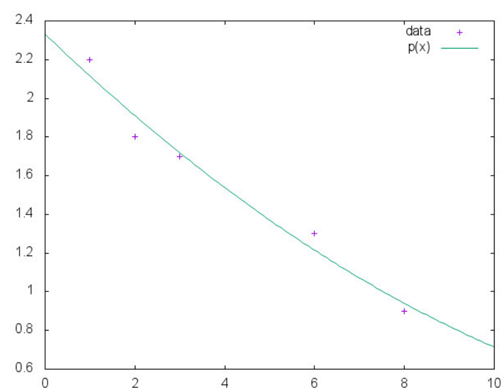
$A\, m \times n, B\, n \times p, C\, m \times p$ i `prodMatVec` (definida a dalt).
Un cop tinguem el sistema d'equacions normals, es resoldrà usant les tècniques i funcions de les dues primeres parts. Caldrà escriure la solució x^* i el valor $\|Ax^* - b\|_2$

Per testejar la funció calculeu les x^* del sistemes següents:

$$A_1 = \begin{pmatrix} 1 & 0.25 \\ 1 & 0.50 \\ 1 & 0.75 \\ 1 & 1.00 \\ 1 & 1.25 \\ 1 & 1.50 \\ 1 & 1.75 \end{pmatrix}, b_1 = \begin{pmatrix} 0.40 \\ 0.50 \\ 0.90 \\ 1.28 \\ 1.60 \\ 1.66 \\ 2.02 \end{pmatrix}; A_2 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 3 & 2 \\ 1 & 2 & 3 \\ 1 & 1 & 4 \end{pmatrix} b_2 = \begin{pmatrix} 5 \\ 9 \\ 12 \\ 16 \end{pmatrix}.$$

Volem trobar un polinomi de segon grau que aproximi les dades de la taula

z_i	1	2	3	6	8
$f(z_i)$	2.2	1.8	1.7	1.3	0.9



per això hem de "resoldre" el sistema sobredeterminat de matriu A (5×3) amb $a_{ij} = z_i^j$ $i = 0, 1, 2, 3, 4$, $j = 0, 1, 2$ i terme independent b amb $b_i = f(z_i)$ $i = 0, 1, 2, 3, 4$. Calculeu la matriu i el vector, doneu les dades al vostre programa i comproveu que el polinomi buscat és
$p(x) = 2.3326945548025217 - 0.22300771208226053\, x + 0.0061112409441456 44\, x^2$
i el residu 1.5421748739693291

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 minúscules (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-1` correspon a una alumna del grup a que fa la pràctica 1.
Aquest directori contindrà els arxius .c corresponents a les diverses parts:

a) Arxiu `prac1funs.c` que conté les funcions `resTinf`, `resTsup`, `prodMatVec`, `prodMatMat`.

b) Arxiu `prac1funs.h` que conté les **capçaleres** de les funcions `resTinf`, `resTsup`, `prodMatVec`, `prodMatMat` per a usar un `#include` en els arxius .c

c) Arxiu `prac1a.c` que conté el programa principal de la primera part.

d) Arxiu `prac1b.c` que conté el programa principal de la segona part i la funció `ldlt`.

e) Arxiu `prac1c.c` que conté el programa principal de la tercera part i la funció `ldlt`.

No es poden incloure funcions en els arxius tret de les que apareixen en el texte

Es crearà un arxiu comprimit del directori amb la comanda

```
tar -czvf a-LopezPerezMaria-1.tgz a-LopezPerezMaria-1
```

executada des del directori pare.

Entregar la pràctica vol dir el següent:

(1) Es penjarà el fitxer comprimit (.tgz) al campus virtual abans del **11 de novembre de 2017**.

Tots els arxius .c lliurats hauran de començar amb les dades de l'alumne en un comentari de la forma

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

Tots els programes hauran de compilar amb les opcions: `-ansi -pedantic -O -Wall`.
Lliurar un programa sense les dades personals (usant l'estil anterior), o amb algun error o avís (warning) de compilació, serà avaluat amb la qualificació mínima.

(2) Entregar un document (imprès) durant l'hora del laboratori del **15 de novembre de 2017** on es doni resposta argumentada a les diferents qüestions que es proposin en relació a la pràctica.