Primera pràctica: descomposició LU

1 Objectius

Programar en llenguatge *C/C++* l'algorisme de la descomposició LU d'una matriu A usant pivotatge parcial esglaonat. Resolució de sistemes lineals llegint les dades des d'un fitxer de text.

2 Comentaris

A continuació es descriuen les funcions que caldrà entregar. Les funcions que heu fet durant el desenvolupament de la pràctica (essencialment de comprovació) no s'han d'entregar si no és que es criden des d'alguna de les funcions que sí cal entregar. Cal que respecteu estrictament tant el nom de les funcions com la passa de paràmetres. Els índexos de tots els vectors i matrius comencen des del zero *(zero-offset)*.

3 Funcions i programes que caldrà entregar

S'hauran de lliurar quatre arxius: `lu.cc`, `resol.cc`, `sistema.cc` i `main.cc`; cadascun d'ells contenint una funció amb el mateix nom i amb l'entrada i sortida que s'indica.

Remarca. Aquests són els **únics** fitxers que es podràn sotmetre. Qualsevol altra funció *que sigui necessària* —en el sentit que s'assenyala dalt a la Secció 2—, s'haurà d'incloure en algun d'ells.

```
int lu(double **a, int n, int perm[], double tol)
```

Objectiu: descomposició LU d'una matriu A usant pivotatge parcial esglaonat.

Input

`a`: apuntador als apuntadors a les files de la matriu A.

`n`: dimensió de la matriu.

`tol`: tolerància admesa sobre els pivots per decidir si la matriu A és singular o no.

Output

`a`: apuntadors als apuntadors a les files de la matriu A que ara es troba descomposada en LU.

`perm`: vector de permutació de la descomposició de la matriu A en LU, *tal com s'ha explicat a teoria.*

`lu`: la funció ha de tornar un d'aquests valors:

- $1 \rightarrow$ La descomposició ha estat exitosa i ha calgut fer un nombre parell de permutacions.
- $-1 \rightarrow$ La descomposició ha estat exitosa i ha calgut fer un nombre senar de permutacions.
- $0 \rightarrow$ La descomposició no ha estat exitosa. Llavors la matriu A és singular d'acord amb la tolerància `tol`. En aquest cas hem destruït la matriu, ja que no tenim ni la matriu A d'entrada ni la seva descomposició LU.

```
void resol(double **a, double x[], double b[], int n, int perm[])
```

Objectiu: Resolució d'un sistema lineal $Ax = b$ on la matriu A, no singular, la tenim factoritzada LU.

Input

`a`: apuntador als apuntadors a les files de la matriu A descomposada LU.

`b`: terme independent del sistema lineal.

`n`: dimensió del sistema.

`perm`: vector de permutació de la descomposició de A en LU.

Output

`x`: solució del sistema lineal $Ax = b$.

```
int sistema(double **a, double x[], double b[], int n, double tol)
```

Objectiu: Resolució d'un únic sistema usant les funcions `lu` i `resol`. Alocatació dinàmica dels vectors de treball que calguin.

Input

`a`: (com a la funció `lu`) apuntador als apuntadors a les files de la matriu A.

`n`: (com a la funció `lu`) dimensió de la matriu.

`tol`: (com a la funció `lu`) tolerància admesa sobre els pivots per decidir si la matriu A és singular o no.

`b`: (com a la funció `resol`) terme independent del sistema lineal.

Output

`a`: apuntador als apuntadors a files de la matriu A que ara està descomposada LU però que de fet la podem considerar *destruïda* ja que no disposem de la permutació.

`x`: solució del sistema $Ax = b$.

`sistema`: tal com l'output `lu` de la funció `lu`, i.e.,

- $1 \rightarrow$ La descomposició ha estat exitosa i ha calgut fer un nombre parell de permutacions.
- $-1 \rightarrow$ La descomposició ha estat exitosa i ha calgut un nombre senar de permutacions.
- $0 \rightarrow$ La descomposició no ha estat exitosa. Llavors la matriu A és singular d'acord amb la tolerància `tol`. En aquest cas hem destruït la matriu, ja que no tenim ni la matriu A d'entrada ni la seva descomposició LU.

```
int main(int argc, char *argv[])
```

Objectiu: Programa que llegeix un sistema lineal d'un arxiu, el resol i escriu la solució en un altre arxiu. S'han de fer alocatacions dinàmiques de memòria per a tots els vectors i matrius que calguin.

Input L'arxiu d'entrada serà un arxiu ASCII —que s'haurà de llegir des de la línia de comandes—, contenint els termes no nuls del sistema lineal amb el següent format: (n, m, k, i_s, j_s, i_r representen enters i $a_{i_s j_s}, b_{i_r}$ reals doubles, amb $s = 1, \ldots, m$, $r = 1, \ldots, k$).

n			dimensió del sistema (n)
m			nombre de components no nul·les d'A (m)
i_1	j_1	$a_{i_1 j_1}$	
i_2	j_2	$a_{i_2 j_2}$	
$\vdots$	$\vdots$	$\vdots$	m files
i_m	j_m	$a_{i_m j_m}$	
k			nombre de components no nul·les de b (k)
i_1	b_{i_1}		
i_2	b_{i_2}		
$\vdots$	$\vdots$		k files
i_k	b_{i_k}		

Taula 1

Output Un arxiu ASCII contenint la solució amb el següent format (les x_i amb $i = 0, \ldots, n-1$ són reals doubles).

0	x_0
1	x_1
$\vdots$	$\vdots$
$n-1$	x_{n-1}

Taula 2

Remarca. Notem que si volem conèixer com de bona és la solució obtinguda, podem calcular la corresponent norma del residu, que definim com: `res2` $:= \|Ax - b\|_2$, o bé `resInf` $:= \|Ax - b\|_\infty$.