MÈTODES NUMÈRICS I

Grau de Matemàtiques. Curs 2016/17. Semestre de tardor

PRÀCTICA 5

Exercici 1 [Factorització PA=LU]

Feu una funció de prototipus

```
int palu(int n, double **A, int *p, double tol);
```

que calculi la factorització `PA=LU` d'una matriu quadrada A usant **eliminació gaussiana amb pivotatge parcial**. Els paràmetres són:

- `n`: Dimensió de A.
- `A`: En la invocació, conté A. A la sortida, els elements essencials de L i U.
- `p`: Vector de n components enteres que, a la sortida, conté una permutació del conjunt $\{0, 1, 2, \ldots, n-1\}$, la qual representa la matriu P en el sentit següent:

  $\forall i = 0, 1, \ldots, n-1$, la fila i de PA és la fila $p[i]$ de A.
- `tol`: Tolerància per a decidir si un pivot de l'eliminació es considera, o no, zero.

La funció retorna:

- o bé el signe de la permutació p final (± 1), si s'ha pogut fer la factorització;
- o bé 0, si no s'ha pogut fer la factorització.

Completeu l'exercici amb un programa `main` on es llegeix `n`, es gestiona la memòria, es llegeix A, es fixa la tolerància (per ex. 10^{-10}), s'invoca la funció `palu` i s'escriu, o bé el resultat (elements essencials de L i de U, i vector p), o bé algun missatge d'error.

Com a comprovació, es pot calcular i escriure el producte LU.

Exercici 2 [Temps d'execució]

Feu un programa que sigui una modificació de l'anterior i compleixi:

- La matriu A és aleatòria, amb tots els elements a l'interval $[-1, +1]$. Useu les funcions `srand()` i `rand()` i la constant `RAND_MAX`.
- Es calcula el temps d'execució de la funció `palu`. Consulteu la referència

  `http://www.gnu.org/software/libc/manual/html_node/CPU-Time.html`

  Per tal que aquest temps sigui comptabilitzable, cal que n sigui moderadament gran (de l'ordre de milers). Per tant, no feu escriure cap matriu ni vector.

Comproveu si es verifica que el temps d'execució és aproximadament proporcional a n^3.

Exercici 3 [Sistemes tridiagonals]

Feu una funció de prototipus

```
int d3(int n, double *a, double *b, double *c, double *f, double tol);
```

on es resolgui un *sistema tridiagonal* $Ax = f$ de dimensió$n \times n$ mitjançant:

- Factorització LU de la matriu A.
- Resolució del sistema $Ly = f$ per substitució endavant.
- Resolució del sistema $Ux = y$ per substitució endarrera.

La matriu A està donada per 3 vectors `a`, `b` i `c`, mentre que les matrius buscades L i U, estan determinades per un i per dos vectors, respectivament. S'han d'usar els mateixos vectors `a`, `b` i `c`,

També el vector y, i després el vector x, poden ser el mateix vector inicial `f`.

Si, en algun moment, cal fer una divisió per una quantitat de valor absolut menor que `tol`, llavors no es continua i es retorna un valor diferent de 0. Si es poden fer tots els càlculs, la funció retornarà un 0.

Completeu el programa amb una funció `main` on es llegeix `n`, es guarda memòria per als 4 vectors, es llegeixen aquests vectors, es fixa la tolerància, s'invoca la matriu i, finalment, s'escriu, o bé el vector solució, o bé algun missatge d'error.