Práctica 4
Librería LAPACK

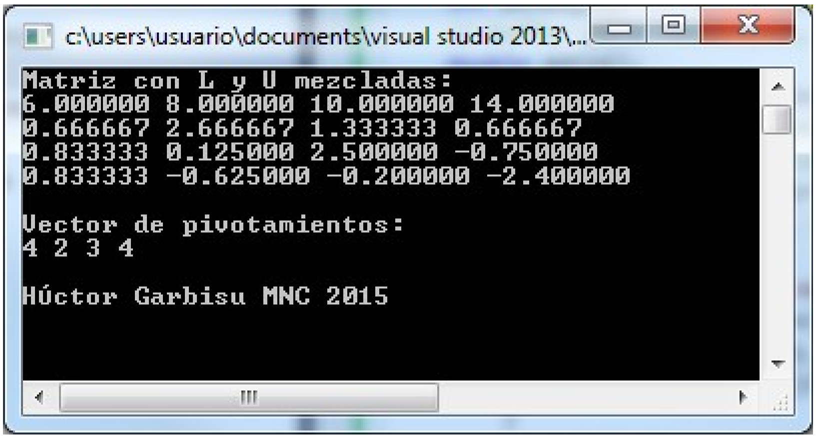
Héctor Garbisu Arocha
Curso 2015/16
Métodos Numéricos para la Computación
Grado en Ingeniería Informática
Escuela de Ingeniería Informática
Universidad de Las Palmas de Gran Canaria

Índice

1. Tarea 1. Descomposición LU. Nivel computacional.

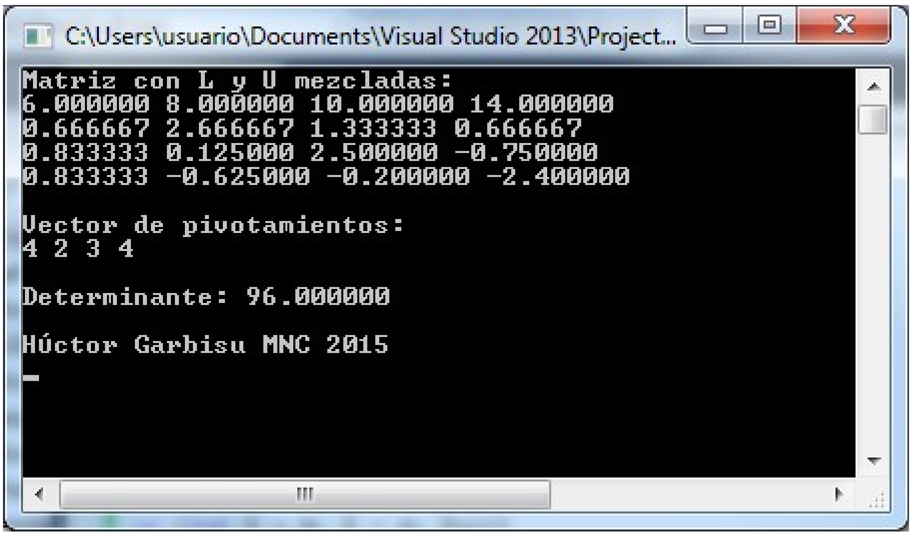
1.1. Fase 1. Obtener LU e imprimir

La función de LAPACK que hace la descomposición LU, a diferencia de la de Matlab, devuelve un vector de pivotamiento (en contraposición a una matriz de pivotamiento) y una sola matriz que representa las dos matrices triangulares L y U. Sabiendo que la diagonal de L son todo 1s no se pierde información al guardar ambas matrices triangulares en una compuesta: A2; ya vamos viendo que en LAPACK al igual que en BLAS todo es por la eficiencia.



1.2. Fase 2. Computo del determinante

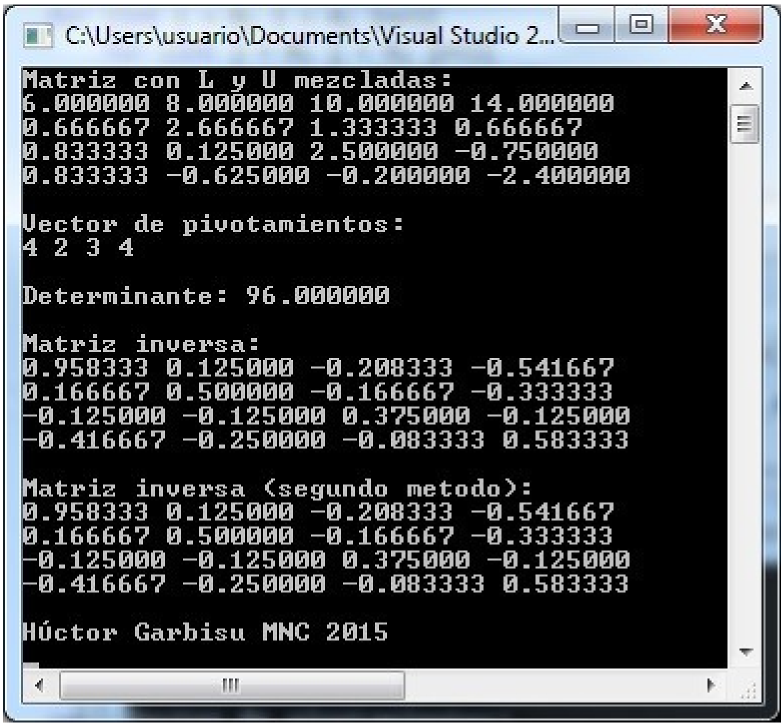
Antes de pedirle a LAPACK que invierta la matriz LU debemos comprobar que sea invertible. Para ello demostramos que el determinante no es 0 usando una de las propiedades de las matrices triangulares, que permite obtener el determinante como una función del producto de sus elementos diagonales. Como la diagonal L son todo unos, el determinante de A se podrá calcular con sólo unas multiplicaciones en la diagonal de A2.



1.3. Fase 3. Calculo de la matriz inversa

Ya sabemos que la matriz es invertible y vamos a usar dos métodos diferentes para obtener la inversa.
El primer cómputo consistirá en calcular X en A * X = B , donde B es la matriz identidad y A es la matriz A2.
Este método no sobreescribe la matriz LU porque da el resultado en otra, la que se ha pasado como matriz identidad.

El segundo método es aparentemente más directo. Se usa la función LAPACL_dgetri que computa la matriz inversa de LU a partir de LU y el vector de pivotamiento.



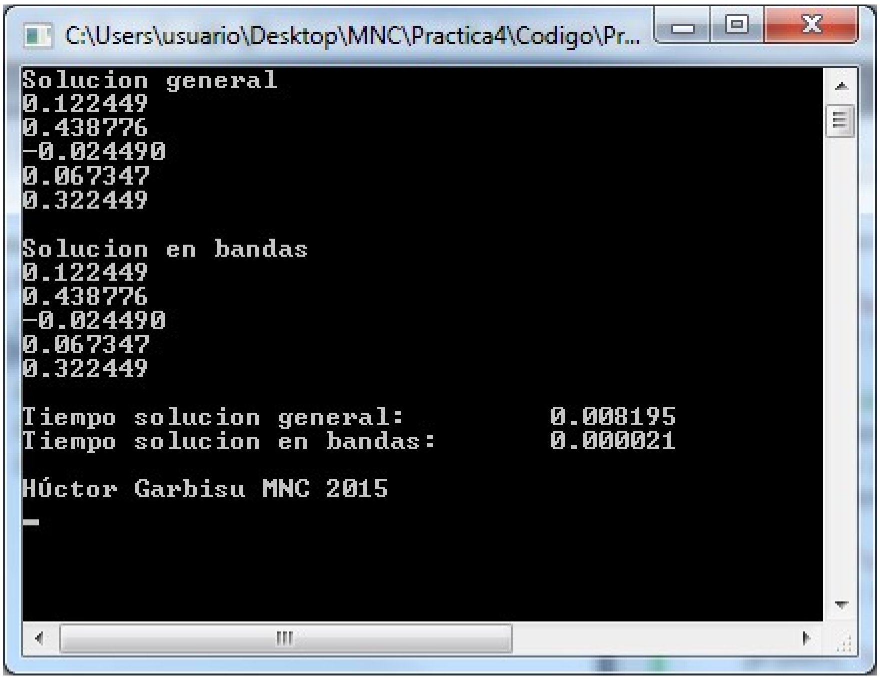
A algo en el proceso le ha hecho gracia cambiarme el nombre al no distinguir entre 'é' y 'ú'.

2. Tarea 2. Resolver Sistema Lineal. Nivel Driver.

Vamos a comparar la eficiencia de la librería para resolver sistemas de ecuaciones con las funciones para matrices generales y otra para matrices en bandas.

Con el fin de hacer una demostración más o menos clara , vamos a usar la misma matriz de ecuaciones, primero escrita y resuelta como matriz general y luego usando la notación y algoritmo para matrices en banda.

El primer resultado es escrito en la matriz X mediante la función LAPACKE_dgesv.
El segundo, es el resultado de la función LAPACKE_dgbsv.



Como se puede ver, no hay ninguna diferencia en el resultado, mientras que el tiempo de ejecución es menor para la función que optimiza el cálculo para matrices en banda.

A pesar de que este es un caso muy pequeño (5x5), no sería de extrañar que esta ganancia en eficiencia fuera también considerable para sistemas de ecuaciones mucho mayores, ya que la resolución de matrices en banda es muy fácil.