Práctica 3 Métodos numéricos

Francisco Javier Blázquez Martínez

Resumen

Trabajo realizado en el marco de la asignatura *Métodos numéricos*. Explicación del código entregado de la práctica de resolución de sistemas lineales por métodos directos. Contiene código desarrollado en *MATLAB* para la factorización de Cholesky y LU de matrices y resolución de sistemas mediante el método de eliminación gaussiana entre otros. Se incluyen ejemplos de ejecución para aclarar dudas en el testeo de los métodos.

1. Sobre la entrega

He optado por segmentar mi código para hacerlo más modular y más fácilmente testeable y a la vez separar así lo que era la pura funcionalidad de las factorizaciones de lo que es la interfaz de uso de estas. Es por esto que los apartados de esta práctica no los he generado en un único script de MATLAB y me es por tanto imposible subirlos a la entrega del campus virtual de otra forma que no sea comprimidos (formato .zip). He decidido por tanto por subir un .zip con todos los ficheros que tienen dependencias en las entregas. En otras palabras, de las cuatro entregas abiertas, he subido en las cuatro el mismo .zip con la funcionalidad requerida por todos los apartados de la entrega para que así sea tan sencillo como descomprimir este .zip y abir la carpeta desde MATLAB para probar la funcionalidad de cualquier apartado.

2. Factorización LU

- **Script de ejecución:** FactorizacionLU.m
- **Código factorización:** LUfactorization.m
- **Otras dependencias:** lowerTriangularSystem.m, upperTriangularSystem.m

Creo que el código está bien explicado y es claro. He prescindido de bucles for anidados para mejorar el rendimiento. Incluyo el ejemplo 4.3 del libro a continuación:

```
>> FactorizacionLU
------------------------------------------------------------
Resolucion del sistema Ax=b hallando la factorizacion LU de A
------------------------------------------------------------
Introduzca la matriz A: [1 2 1 3; 1 1 1 4; 2 1 4 10; -1 -3 7 5]
Introduzca el termino independiente b: [45; 48; 101; -4]
L =
     1     0     0     0
     1     1     0     0
     2     3     1     0
    -1     1     4     1

U =
     1     2     1     3
     0    -1     0     1
     0     0     2     1
     0     0     0     3

x =
     5
     7
    -4
    10

Quiere variar el vector de terminos independientes? [SI=1/NO=0]: 1
Introduzca el nuevo termino independiente b: [0;0;1;2]

El nuevo valor de x es:
    2.5000
   -0.6667
    0.8333
   -0.6667

Quiere variar el vector de terminos independientes? [SI=1/NO=0]: 0
>>
```

3. Factorización de Cholesky

- **Script de ejecución:** FactorizacionCHOL.m
- **Código factorización:** CHOLfactorization.m
- **Otras dependencias:** lowerTriangularSystem.m, upperTriangularSystem.m

En este apartado, de nuevo al no manejar punteros gracias a lo restrictivas que son las condiciones para que se pueda dar la factorización de Cholesky, el código con los comentarios contenidos debe ser sencillo de entender. Lo único es que tiendo a programar en inglés y las funciones tienen los comentarios en este idioma. Mis disculpas, es una manía mía y me ha dado una gran pereza ponerme a cambiarlo ahora. Incluyo como ejemplo de ejecución el ejemplo resuelto 4.4 del libro.

```
>> FactorizacionCHOL
------------------------------------------------------------
Resolucion del sistema Ax=b hallando la factorizacion Cholesky de A
------------------------------------------------------------
Introduzca la matriz A: [1 -1 1 0; -1 2 -1 2; 1 -1 5 2; 0 2 2 6]
Introduzca el termino independiente b: [4; -3; 16; 8]
B =
     1     0     0     0
    -1     1     0     0
     1     0     2     0
     0     2     1     1

x =
     2
     1
     3
     0

Quiere variar el vector de terminos independientes? [SI=1/NO=0]: 0
>>
```

4. Eliminación gaussiana

- **Script de ejecución:** EliminacionGauss.m
- **Código factorización:** PALUfactorization.m
- **Otras dependencias:** lowerTriangularSystem.m, upperTriangularSystem.m

Para este apartado, empleamos la eliminación gaussiana para hallar la factorización PA=LU sobre la propia matriz A tal y como se ve en el apartado 4.3.4 del libro. Sin embargo, el método calcula L y U sobre la propia matriz A pero después las genera para devolver estas explícitamente. Las permutaciones de filas de la matriz se realizan gracias a un vector de punteros para reducir el número de operaciones a ejecutar. Añado el ejemplo de ejecución de mi implementación con el ejemplo 4.2 del libro.

```
>> EliminacionGauss
------------------------------------------------------
Resolucion del sistema Ax=b por eliminacion gaussiana
------------------------------------------------------
Introduzca la matriz A: [0 1 2 1; 1 2 1 3; 1 1 -1 1; 0 1 8 12]
Introduzca el termino independiente b: [1; 0; 5; 2]
La matriz A admite factorizacion PA=LU de la forma:
L =
     1     0     0     0
     0     1     0     0
     0     1     1     0
     1    -1     0     1

U =
     1     2     1     3
     0     1     2     1
     0     0     6    11
     0     0     0    -1

El vector de permutacion de filas es:
     2     1     4     3

x =
   37.5000
  -15.3333
   11.1667
   -6.0000

Quiere variar el vector de terminos independientes? [SI=1/NO=0]: 1
Introduzca el nuevo termino independiente b: [1;1;1;1]
El nuevo valor de x es:
    5.5000
   -1.6667
    1.8333
   -1.0000

Quiere variar el vector de terminos independientes? [SI=1/NO=0]: 0
>>
```

5. Resolución de sistemas tridiagonales

- **Script de ejecución:** Tridiagonales.m
- **Código factorización:** tridiagonalSystem.m

En este apartado cobra especial importancia el ejemplo de cómo he decidido que debía hacerse la entrada de datos por consola. Al querer resolver un sistema con una matriz de coeficientes nula, salvo por su diagonal principal y sus dos diagonales contiguas, he optado por leer únicamente los datos de estas diagonales en el siguiente orden. Primero la diagonal inferior, luego la principal y tras esta la diagonal superior. He optado por hacerlo así porque es el orden que se le da en el ejercicio 7 de la hoja 3. Además, para mayor agilidad las diagonales se leen como un vector fila mientras que el término independiente del sistema se tiene que seguir leyendo como un vector columna. Incluyo un ejemplo de ejecución propio muy sencillo.

```
>> Tridiagonales
-----------------------------------------------------------
Resolucion del sistema Ax=b siendo A una matriz tridiagonal
-----------------------------------------------------------
Introduzca la diagonal inferior  de A: [1 1]
Introduzca la diagonal principal de A: [1 1 2]
Introduzca la diagonal superior  de A: [0 0]
Introduzca el termino independiente b: [1; 2; 3]
A =
     1     0     0
     1     1     0
     0     1     2

x =
     1
     1
     1

>>
```

Referencias

[1] Juan Antonio Infante del Río y José María Rey Cabezas. *Métodos numéricos. Teoría, problemas y prácticas con MATLAB.* Editorial pirámide, 4ª edición. Libro de la asignatura “Métodos numéricos”,Facultad de ciencias matemáticas, Universidad Complutense, Madrid, curso 2018/19.