UNIVERSIDAD
DE MURCIA

DEPARTAMENTO DE MATEMÁTICAS

MÉTODOS NUMÉRICOS. PRÁCTICA 4

Iniciamos una serie de prácticas sobre matrices y sistemas lineales que iremos guardando en nuestro proyecto `MetodosNumericos` en paquetes con el nombre `metodosLineales.practicaX` (en particular el correspondiente a esta práctica es el `metodosLineales.practica4`). De momento hemos añadido la clase `Matrices.java` al paquete `auxiliar`. En esta clase guardaremos a lo largo de esta parte de las prácticas todos nuestros métodos de manipulación de matrices, resolución de sistemas lineales y cálculo de valores y vectores propios, pero por lo pronto sólo incluye la constante `precision`, que es la que usaremos en la clase para determinar cuando un número es o no cero, los métodos `ifMatriz` e `ifCuadrada` para saber si un objeto `double[][]` es una matriz (es decir, todas las filas tienen la misma longitud) o una matriz cuadrada, respectivamente, dos variantes del método `suma` que permiten sumar dos matrices o vectores de las mismas dimensiones, y una extension de la clase `Exception` (`ErrorMetodosLineales`) que utilizaremos cuando lancemos errores desde los métodos de la clase.

1. Diseñar diversos métodos `resta` y `producto` para realizar el resto de las operaciones habituales de vectores y matrices. Por ejemplo

```
double[] producto(double[][] A, double[] v)
```

calcularía el producto de la matriz A por el vector v,

```
double producto(double[] u, double[] v)
```

el producto (escalar) de los vectores u y v y

```
double[][] producto(double[][] A, double[][] B)
```

el producto de las matrices A y B. Asimismo, diseñar sendos métodos `toString` para escribir matrices y vectores, un método `double[][] traspuesta (double[][] A)` para calcular la traspuesta de una matriz A y un método `double norma(double[] v)` para calcular la norma euclídea de un vector v.

2. Consideremos las matrices

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}, \quad A' = \begin{pmatrix} 10 & 7 & 8.1 & 7.2 \\ 7.08 & 5.04 & 6 & 5 \\ 8 & 5.98 & 9.89 & 9 \\ 6.99 & 4.99 & 9 & 9.98 \end{pmatrix}, \quad A^{-1} = \begin{pmatrix} 25 & -41 & 10 & -6 \\ -41 & 68 & -17 & 10 \\ 10 & -17 & 5 & -3 \\ -6 & 10 & -3 & 2 \end{pmatrix}$$

y los vectores

$$b = (32, 23, 33, 31), \quad b' = (32.1, 22.9, 33.1, 30.9).$$

Comprobar que A^{-1} es la inversa de A y verificar, calculando las normas de los vectores $Au - b$, $A'v - b$ y $Aw - b'$, que

$$u = (1, 1, 1, 1), \quad v = (-81, 137, -34, 22), \quad w = (9.2, -12.6, 4.5, -1.1)$$

son soluciones respectivas de los sistemas $Ax = b$, $A'x = b$ y $Ax = b'$.

3. Escribir un método recursivo

```
double determinanteMenores(double[][] A)
```

que calcule el determinante de una matriz cuadrada de números reales desarrollando por menores complementarios. Como aplicación, considérese la matriz cuadrada tridiagonal de dimensión n

$$A_n = \begin{pmatrix} 0 & 1 & & & & \\ 1 & 0 & 1 & & & \\ & 1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 0 & 1 \\ & & & & 1 & 0 \end{pmatrix}$$

y calcúlese su determinante para $n = 2, 3, \ldots, 10$. ¿Qué ocurre desde $n = 11$ en adelante?

4. Añadir a la clase `Matrices` un método `norma1` que calcule $\|A\|_1$ para una matriz cuadrada A, y a partir de éste un método `aproxRadioEspectral` que aproxime el radio espectral de una matriz cuadrada utilizando el teorema que afirma que si $\|\cdot\|$ es una cierta norma matricial entonces $\lim_{k\to\infty} \|A^k\|^{1/k} = \rho(A)$. (NOTA: como el procedimiento descrito en el teorema es un límite que puede ser inestable, usar como condición de parada el que $\|A^k\|$ no sea un número de máquina (`NaN`) o que sea `Infinity`.)

Utilizando el método anterior y las matrices del ejercicio 2, calcular `aproxRadioEspectral(`A`)` y `aproxRadioEspectral(`A^{-1}`)` y usar estos números para dar una cota inferior de $\mathrm{cond}_2(A)$. Calcular también $\mathrm{cond}_1(A)$. A la luz de estos valores, ¿qué cabe concluir acerca de los resultados del ejercicio 2?

5. Crear un método `double[] reglaCramer(double[][] A, double[] b)` para resolver un sistema $Ax = b$ mediante la regla de Cramer (usar `determinanteMenores` para calcular los correspondientes determinantes). El método se guardará en la clase `Matrices` y lanzará los pertinentes errores. Verificar su eficacia resolviendo el sistema

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}$$

en una clase ejecutable `Ejercicio5`.

6. Es conocido que para cada matriz cuadrada A las series matriciales

$$\sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} A^{2k}$$

y

$$\sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} A^{2k+1}$$

convergen. Las matrices suma de dichas series se denominan, respectivamente, el *coseno* y el *seno* de la matriz A. Añadir a la clase `Matrices` sendos métodos `cos` y `sen` que reciban una matriz cuadrada A y un entero r y devuelvan, respectivamente,

$$\sum_{k=0}^{r} \frac{(-1)^k}{(2k)!} A^{2k}$$

y

$$\sum_{k=0}^{r} \frac{(-1)^k}{(2k+1)!} A^{2k+1}$$

Comprobar su eficacia aplicándolos a la matriz

$$A = \begin{pmatrix} 2 & 3 & -1 \\ -3 & 5 & -8 \\ 0 & 5 & 7 \end{pmatrix}$$

y verificando la validez de la fórmula matricial

`cos(A)cos(A) + sen(A)sen(A) = Id.`