UNIVERSIDAD
DE MURCIA

Departamento de Matemáticas

Métodos Numéricos. Práctica 8

1. Construir un método

```
double[] potencia(double[][] A, double tol, int nmax,
    double[] v, double[] y)
```

que, a partir de una matriz A de dimensión n y aplicando el método de la potencia devuelva un vector de dimensión $n+1$ cuya última coordenada es el valor propio dominante de A y cuyas primeras n coordenadas son las del correspondiente vector propio.

2. Usando el método anterior, construir un método

```
double[][] deflacionWielandt (double[][] A, double tol, int nmax)
```

que obtenga usando el método de deflación de Wielandt los valores y vectores propios de una matriz A. Si la matriz es de dimensión n, la matriz que devuelva el método será de dimensiones $n \times (n+1)$, constando cada una de las filas del vector propio y el valor propio correspondiente. Verificarlo con la matriz

$$A = \begin{pmatrix} 4 & -1 & 1 \\ -1 & 3 & -2 \\ 1 & -2 & 3 \end{pmatrix}.$$

3. Sabiendo que los valores propios de la matriz

$$A = \begin{pmatrix} -2 & 0 & 1 & 1 \\ -5 & -5 & 4 & 5 \\ 16 & 14 & -8 & -12 \\ -19 & -15 & 11 & 15 \end{pmatrix}$$

son de la forma $-\lambda_1 < -\lambda_2 < 0 < \lambda_2 < \lambda_1$, usar una variante adecuada del método de la potencia para calcularlos y encontrar una base de vectores propios.

4. Construir un método

```
double[][] jacobiPropios(double[][] A, double tol, int nmax)
```

que obtenga los valores y vectores propios de una matriz simétrica A usando el método de Jacobi.

5. Construir una aplicación que verifique la efectividad del método anterior cuando se aplica a las matrices

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

6. Comparar la efectividad de los métodos `deflacionWielandt` y `jacobiPropios` cuando se aplica a la matriz A de coeficientes $a_{ij} = i + j + 1$, $0 \le i, j \le 14$.

7. Constrúyase una variante `potenciaMasProxima` del método `potencia` (como éste se guardará en `Matrices.java`) que, a partir de una matriz cuadrada dada A y de un cierto número λ_0, proporcione el valor propio de A más cercano a λ_0 y el correspondiente vector propio para A. Verifíquese con $\lambda_0 = 3$ y

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 3 & 1 \\ 1 & 1 & 1 & 4 \end{pmatrix}.$$