4 Cálculo de autovalores y autovectores

4.1 El método de la potencia

El método de la potencia permite obtener una aproximación del autovalor de módulo máximo y de un autovector asociado.

La idea de este método radica en que, si A tiene un autovalor λ cuyo módulo sea estrictamente superior al de todos los demás y x es un vector "cualquiera", entonces, al multiplicar sucesivamente por la matriz A, la componente de $A^k x$ en el espacio propio $V_\lambda(A)$ asociado al autovalor λ tenderá a hacerse "dominante" y se tendrá

$$\lim_{k\to\infty} A^k x = v \in V_\lambda(A).$$

Sea $A \in \mathbb{C}^{n\times n}$ diagonalizable y tal que tiene un autovalor de módulo máximo (no tiene que ser simple). Suponemos que los autovalores de A están ordenados de la forma

$$|\lambda| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_m| \quad \text{con } m \leq n.$$

Algoritmo (método de la potencia)
Entrada: $A \in \mathbb{R}^{n\times n}$; *tol* tolerancia para el test de parada; $Imax$ número máximo de iteraciones
Salida: λ autovalor de módulo máximo de A y $v \in V_\lambda(A)$

a) Elegir $u^0 \in \mathbb{R}^n$, hacer $\sigma^0 = 0$. Elegir una aplicación lineal $\phi\colon \mathbb{R}^n \to \mathbb{R}$ (tipicamente, $\phi(v) = v_j$ para algún $1 \leq j \leq n$).

b) Dados $k \geq 0$ y $u^k \in \mathbb{R}^n$, repetir los cálculos siguientes, hasta que sea $|\sigma^{k+1} - \sigma^k| < tol$ o bien hasta que se alcancen $Imax$ iteraciones:

- **b.1)** Calcular $w^k = \dfrac{u^k}{\|u^k\|}$
- **b.2)** Calcular $u^{k+1} = Aw^k$
- **b.3)** Calcular $\sigma^{k+1} = \dfrac{\phi(u^{k+1})}{\phi(w^k)} = \dfrac{u_j^{k+1}}{w_j^k}$.

c) Cuando se obtenga la convergencia, hacer $\lambda = \sigma^k$ y $v = u^{k+1}$ (o bien $v = \dfrac{u^{k+1}}{\|u^{k+1}\|}$, si se quiere un vector propio normalizado).

4.2 Cálculo de autovalores con MATLAB

Para calcular los autovalores de una matriz, MATLAB dispone de la orden

```
aval = eig(A)
```

que devuelve un vector que contiene **todos** los valores propios de la matriz A.
Si se desean calcular también los vectores propios, entonces hay que usar esta orden en la forma

```
[V, D] = eig(A)
```

que devuelve una matriz diagonal `D` con los valores propios y una matriz `V` cuyas columnas son los correspondientes vectores propios.

4.3 Ejercicios

Ejercicio 4.1 Escribir una M-función para implementar el método de la potencia:
`function [lambda, v, flag] = MPotencia(A, tol, Itermax)`
Argumentos de entrada:
`A`, la matriz A
`tol`, la tolerancia *tol* para el test de parada
`Itermax`, el número máximo de iteraciones a realizar
Argumentos de salida:
`lambda`, la aproximación del autovalor
`v`, la aproximación del vector propio asociado
`flag`, un indicador que valdrá 0 si el algoritmo no converge y el número de iteraciones realizadas si lo hace.

Para verificar el funcionamiento del programa se podrán utilizar las matrices almacenadas en los ficheros de datos siguientes (que se pueden cargar en memoria con la orden `A = load('nombrefichero')`):

`autoval1.txt`: matriz de dimension 30 con un autovalor singular dominante.

`autoval2.txt`: matriz de dimension 30 que no tiene un autovalor dominante.

`autoval3.txt`: matriz simétrica definida positiva de dimension 50 con un autovalor dominante de módulo sensiblemente mayor que el resto, lo cual acelera la convergencia.

`autoval4.txt`: matriz simétrica definida positiva de dimension 50 con un autovalor dominante, pero de magnitud no muy superior a los demás, por lo que la convergencia será más lenta.

Comparar, en todos los casos, con los resultados obtenidos mediante la función `eig` de MATLAB.