

## PROGETTO 2: MOLTEPLICITÀ ALGEBRICA E GEOMETRICA DEGLI AUTOVALORI DI UNA MATRICE

### 1. CONTESTO E FINE

Sia  $A$  una matrice reale  $n \times n$ , gli autovalori di  $A$  hanno due caratterizzazioni:

- $\lambda \in \mathbb{C}$  è autovalore di  $A$  se il sistema  $(A - \lambda \mathbb{I})x = 0$  ammette soluzioni non nulle. In tal caso la dimensione dello spazio vettoriale delle soluzioni del sistema è detta *molteplicità geometrica* dell'autovalore  $\lambda$ . Denotiamo tale quantità con  $\text{mult}_{\text{geo}}(\lambda)$ , dunque

$$(1) \quad \text{mult}_{\text{geo}}(\lambda) := n - \text{rk}(A - \lambda \mathbb{I}).$$

- $\lambda \in \mathbb{C}$  è autovalore di  $A$  se  $\lambda$  è radice dell'equazione  $f_A(x) = 0$ , dove  $f_A(x) := \det(A - \lambda \mathbb{I})$ . In tal caso la molteplicità della radice è detta *molteplicità algebrica* di  $\lambda$ . Denotiamo tale quantità con  $\text{mult}_{\text{alg}}(\lambda)$ , dunque

$$(2) \quad \text{mult}_{\text{alg}}(\lambda) := \{m \in \mathbb{N} : 0 = f_A(\lambda) = f'_A(\lambda) = \dots = f_A^{(m-1)}(\lambda), 0 \neq f_A^{(m)}(\lambda)\}.$$

Una matrice risulta diagonalizzabile, ovvero  $A = P^{-1}\Lambda P$  con  $P$  matrice invertibile e  $\Lambda$  matrice diagonale con gli autovalori sulla diagonale, se e soltanto quando per ogni suo autovalore vale

$$\text{mult}_{\text{geo}}(\lambda) = \text{mult}_{\text{alg}}(\lambda).$$

Vogliamo sviluppare un programma in grado di effettuare un controllo sull'effettiva diagonalizzabilità di una matrice data.

- Per il calcolo della molteplicità geometrica ci basiamo sulla fattorizzazione LU. L'algoritmo LU (con pivoting parziale) produrrà infatti una matrice  $U$  con tante righe "nulle" quanto è la dimensione dello spazio delle soluzioni di  $(A - \lambda \mathbb{I})x = 0$  (perché?). Tale controllo si può effettuare anche sui soli elementi diagonali di  $U$  e va impostata una tolleranza per contare, ad esempio, gli elementi

$$\{i \in \{1, \dots, n\} : |U_{i,i}| < \text{toll}\}.$$

- Per il calcolo della molteplicità algebrica di un autovalore possiamo utilizzare le proprietà di convergenza del metodo di Newton. Sappiamo infatti che il metodo di Newton, qualora volessimo approssimare una radice di ordine  $m$  converge (localmente) con velocità lineare e costante asintotica pari ad  $(m-1)/m$  (perché?), ovvero  $\lim_{k \rightarrow +\infty} |s_k|/|s_{k+1}| = (m-1)/m$ , dove si è denotato con  $s_k$  lo step. Se siamo sufficientemente vicini alla radice avremmo dunque

$$(3) \quad |s_k|/|s_{k+1}| \approx (m-1)/m.$$

### 2. FUNCTION RICHIESTE

**2.1. Calcolo di  $\text{mult}_{\text{geo}}(\lambda)$ .** Vogliamo creare una function `multgeo` avente chiamata `k=multgeo(A,1,toll)` con parametri di input

- **A** matrice quadrata reale o complessa di cui si vogliono calcolare le dimensioni dell'autospazio;
- **1** scalare complesso, è l'autovalore di cui si vuole calcolare la molteplicità algebrica;

- `toll` scalare reale positivo utilizzato per verificare la condizione di nullità degli elementi diagonali della  $U$

ed output

- `k` naturale positivo, dimensione dell'autospazio.

Tale function deve chiamare la fattorizzazione LU di matlab e calcolare quanti elementi diagonali di  $U$  risultano essere in modulo sotto la soglia passata dall'utente.

**2.2. Calcolo di  $f_A(z)$  e  $f'_A(z)$ .** Nell'utilizzo del metodo di Newton necessitiamo del calcolo di  $f_A(z)$  e  $f'_A(z)$  per ogni  $z \in \mathbb{C}$ . Si noti che la formula di Jacobi consente il calcolo di tale derivata, vale infatti la seguente:

$$\frac{d}{dz} \det(B(z)) = \det(B(z)) \operatorname{trace} \left[ B^{-1}(z) \frac{d}{dz} B(z) \right],$$

dove  $\operatorname{trace} B = \sum_{i=1}^n B_{i,i}$ . Notiamo che nel Metodo di Newton  $f'_A$  comparirà solo nella forma  $f_A(z)/f'_A(z)$ , per ragioni di stabilità è dunque meglio calcolare direttamente tale quantità e non  $f_A(z)$  e  $f'_A(z)$  separatamente. Avremo dunque

$$(4) \quad g_A(z) := -\frac{f_A(z)}{f'_A(z)} = \frac{-1}{\operatorname{trace} [(A - z\mathbb{I})^{-1} \frac{d}{dz} (A - z\mathbb{I})]} = \frac{1}{\operatorname{trace} [(A - z\mathbb{I})^{-1}]}.$$

Ricordiamo inoltre che se  $PB = LU$ , dove  $P$  è matrice di permutazione e  $L$  ed  $U$  sono i fattori della fattorizzazione LU, abbiamo

$$\det P \det(B) = \det U = \prod_{i=1}^n U_{i,i}.$$

Inoltre  $\det P = \pm 1$ , dove vale 1 se per portare  $P$  nella matrice identica serve un numero pari di scambi di riga, mentre vale  $-1$  in caso contrario. Quindi possiamo calcolare  $\det P$  con un ciclo sulle colonne che ad ogni passo effettua (se necessario!) uno scambio di righe per portare, al  $j$ -esimo passo, l'unico 1 presente nella  $j$ -esima colonna in posizione  $(j, j)$ . Se il numero di scambi effettuati è  $s$ , allora avremo  $\det P = (-1)^s$ .

Vogliamo implementare una function `myobjective` avente chiamata `[f,g]=myobjective(z,A)` che, per ogni numero complesso  $z$  restituisca  $f_A(z)$  e  $g_A(z)$ .

- Si consideri la fattorizzazione LU con pivoting parziale di  $B(z) := A - z\mathbb{I}$ , si calcoli  $\det(A - z\mathbb{I})$  come  $\det U / \det P$  calcolati come descritto sopra.
- Si calcoli  $g_A(z)$  come descritto nella (4) utilizzando la fattorizzazione LU precedentemente calcolata al fine di invertire la matrice  $B(z)$ .

**2.3. Calcolo di `multalg`  $\lambda$ .** Vogliamo creare una function `multalg` avente chiamata

`[l,m,flag]=multalg(A,l0,toll,it,maxit)` con parametri di input

- `A` matrice quadrata reale o complessa di cui si vogliono calcolare le dimensioni dell'autospazio;
- `l0` punto di partenza del metodo di Newton
- `toll` tolleranza per il test d'arresto dello scarto
- `it` intero maggiore o uguale a 2
- `maxit` intero positivo maggiore di `it`

ed output

- `l` autovalore calcolato
- `m` naturale positivo, molteplicità
- `flag` 1 per successo, 0 per errore

La function dovrà implementare il seguente algoritmo

- Vengano effettuati **it** passi del metodo di Newton partendo da **10**. Se il metodo di Newton è uscito per raggiungimento della tolleranza allora si pone **m**=1 e si esce con **flag**=1, altrimenti si prosegue.
- Venga preso un guess per **m** utilizzando gli ultimi due passi del metodo di Newton e l'euristica (3).
- Si riparta da **10** e si inizializzi il metodo di Newton modificato (i.e.,  $z_{k+1} = z_k - mg_A(z_k)$ ) con al più **maxit** passi
- se il criterio dello step risulta soddisfatto prima di raggiungere il numero massimo di passi consentiti si esce con **flag**=1, altrimenti si incrementi **m** di un unità e si ripeta l'operazione. Se dopo un numero totale di chiamate a **myobjective** di **10maxit** non si è soddisfatto il criterio si esce con **flag**=0.

### 3. SPERIMENTAZIONE

Per sperimentare il programma sviluppato si creino alcuni test "sintetici" e si analizzino i risultati eventualmente producendo grafici esemplificativi.

Evidenziamo qui una procedura per creare diverse matrici test. Possiamo creare matrici con autovalori noti e molteplicità algebriche e geometriche note usando la struttura

$$A = Q^t J Q$$

dove  $Q$  è una matrice ortogonale (per ottenerne una ad esempio prendiamo una matrice random e facciamo la fattorizzazione QR) e  $J$  è una matrice diagonale a blocchi di Jordan. Una tale matrice è definita come matrice diagonale a blocchi

$$J = \begin{pmatrix} J_{k_1}(\lambda_1) & \mathbb{O} & \dots \\ \mathbb{O} & J_{k_2}(\lambda_2) & \dots \\ \vdots & \vdots & \vdots \\ \mathbb{O} & \dots & J_{k_p}(\lambda_p) \end{pmatrix},$$

dove può anche essere  $\lambda_i = \lambda_j$  per indici  $i$  e  $j$  diversi, e dove ciascun blocco diagonale è della forma

$$J_k(\lambda) := \begin{cases} \lambda & k = 1 \\ \begin{pmatrix} \lambda & 1 & 0 & \dots & 0 \\ 0 & \lambda & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \lambda & 1 \\ 0 & \dots & \dots & 0 & \lambda \end{pmatrix} & k > 1 \end{cases} \in M_{k,k}(\mathbb{C})$$

La matrice  $J$  ha allora autovalori  $\lambda_1, \dots, \lambda_p$  aventi molteplicità geometrica

$$\text{mult}_{\text{geo}}(\lambda_i) = \text{Card}\{j \in \{1, 2, \dots, p\} : \lambda_i = \lambda_j\},$$

e molteplicità algebrica

$$\text{mult}_{\text{alg}}(\lambda_i) = \sum_{\{j \in \{1, 2, \dots, p\} : \lambda_i = \lambda_j\}} k_p.$$

Si effettuino test con matrici create ad hoc secondo questa procedura individuando i limiti del software implementato. Si considerino autovalori ben spazziati oppure clusterizzati e di molteplicità bassa (1-2) o alta (3-7).