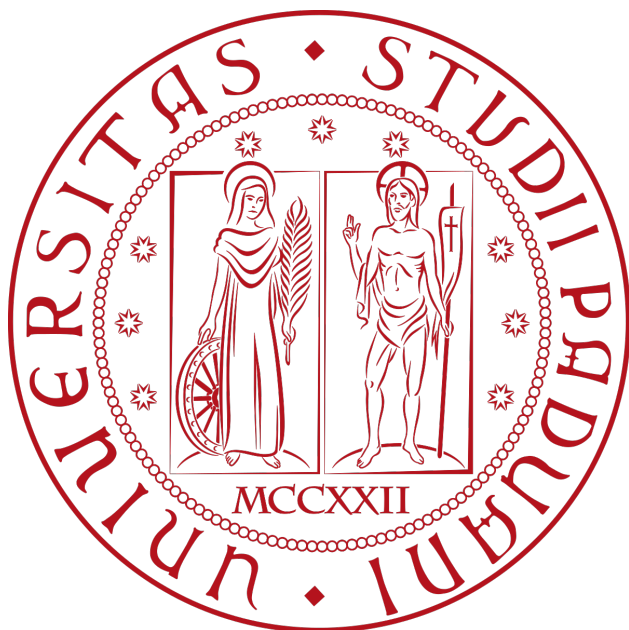


Università degli Studi di Padova



Relazione per: Progetto 2 Matlab

Realizzata da:

Gabriele Di Pietro, matricola 2010000

Michele Dioli, matricola 2077629

Riccardo Berengan, matricola 2080041

1 Introduzione

Gli autovalori sono le soluzioni dell'equazione $\det(A - \lambda I) = 0$. L'autospazio è il sottospazio vettoriale generato da tutti i vettori autovalori λ , insieme al vettore nullo.

Esempio:

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

$$\det(A - \lambda I) = \det \begin{bmatrix} 4 - \lambda & 2 \\ 1 & 3 - \lambda \end{bmatrix} \iff (\lambda - 5)(\lambda - 2) = 0 \iff \lambda_1 = 5, \lambda_2 = 2$$

Molteplicità geometrica è la dimensione dell'autospazio associato ad un autovalore quindi $|n - \text{rk}(A - \lambda I)|$. Molteplicità algebrica è la molteplicità dell'autovalore quindi $|n - \text{rk}(A - \lambda I)|$.

2 Calcolo

2.1 Funzione Multigeo

Per il calcolo contiamo gli elementi "nulli" (*minori di una tolleranza*) sulla matrice U ottenuta dalla fattorizzazione LU . Lo si può fare perché: Una riga nulla in U significa che il sistema lineare di $(A - \lambda I)x = 0$ ha soluzione libera, una riga nulla non dipende dalle altre righe della matrice, le righe nulle indicano che ci sono dipendenze lineari tra le righe. Contiamo quindi gli elementi in diagonale che sono maggiori di una tolleranza, contando così il rango della matrice, lo sottraiamo alla dimensione della matrice e otteniamo la dimensione dell'autospazio dell'autovalore che coincide con la molteplicità geometrica.

2.2 Funzione myobjective

La funzione `myobjective` calcola la funzione f e la funzione g a partire da uno scalare z e da una matrice A quadrata.

Come prima cosa calcoliamo la matrice $B = A - zI$ e poi tramite fattorizzazione LU usando `lu(B)` in Matlab che restituisce la matrice U triangolare superiore e la matrice L triangolare inferiore ed infine la matrice P di permutazione.

$P = \pm 1$ in base al numero di permutazioni effettuate, da qui calcoliamo il determinante di P . Trovato questo calcoliamo il determinante di B come $\det(P) \cdot \prod_{i=1}^n U_{ii}$, dove U_{ii} sono gli elementi sulla diagonale di U . $f = \det(B)$

La funzione dopodiché calcola l'inverso della matrice B e calcola g come l'inverso della traccia (*somma degli elementi sulla diagonale*) di B .

2.3 Funzione Multialg

Consideriamo il polinomio con m molteplicità algebrica $f(x) = (x - \lambda)^m g(x)$ e il metodo di Newton $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ per trovare le radici di $g(x)$. Analizziamo la velocità di convergenza di $f(x) = (x - \lambda)^{m-1}g(x) + (x - \lambda)^m g'(x)$

$f'(x) = 0 \iff x = \lambda$, $m \geq 1$ non è subito alla radice $m(x - \lambda)^{m-1}$ è responsabile per la velocità di convergenza, quindi il metodo di Newton applicato ad una radice di molteplicità $m \geq 1$ ha una velocità lineare non più quadratica.

Abbiamo che la stima dell'errore è ridotta dipendentemente dalla molteplicità della radice

$$|x_{n+1} - \lambda| \approx |x_n - \lambda|^{\frac{m-1}{m}}$$

$\frac{m-1}{m}$ è la velocità di convergenza

$$(Newton\ vicino\ a\ radici\ multiple \Rightarrow |e_{n+1}| \approx C|e_n|^{\frac{m-1}{m}})$$

Calcoliamo l'autovalore con il metodo di Newton, iteriamo fino a un numero di iterazioni e/o per aver raggiunto una tolleranza sensata. Appliciamo Newton al polinomio caratteristico $f(\lambda) = \det(A - \lambda I) \Rightarrow \lambda_{n+1} = \lambda_n - \frac{f(\lambda_n)}{f'(\lambda_n)}$. Troviamo così il valore dell'autovalore.

Per calcolare la molteplicità algebrica basta dividere l'autovalore per il polinomio caratteristico, e vedere quante volte lo posso fare prima di ottenere un valore di zero.

3 Test

3.1 File main.m

Il main permette di definire una matrice di test tramite la modifica del vettore `lambda` e di modificare il punto iniziale per il metodo di newton `10` e il parametro della tolleranza `toll`.

La matrice per i test viene creata tramite la modifica del parametro `lambda` da qui viene invocata una funzione `creaJacob` che prende in input `lambda` e restituisce la matrice in forma diagonale a blocchi di Jordan, poi viene creata una matrice Q tramite `Q = orth(randn(n))` che genera una matrice ortogonale, ed infine viene calcolata la matrice A usando la struttura $A = Q^t J Q$

3.1.1 Test 1