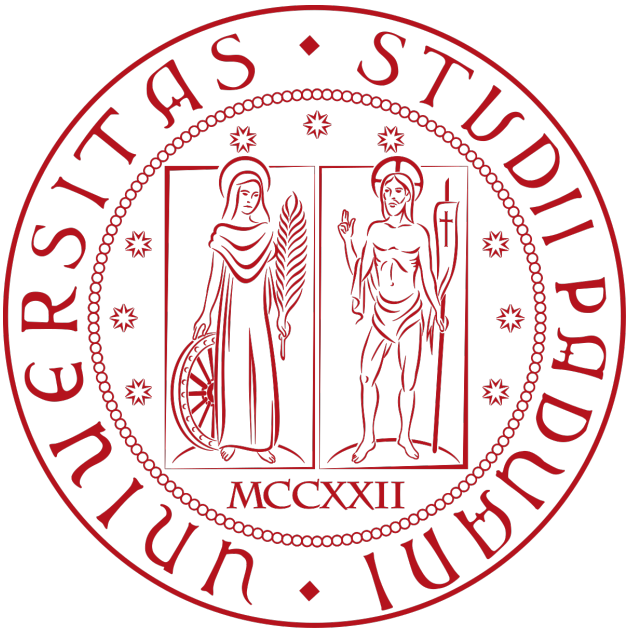
Università degli Studi di Padova



Relazione per:

Progetto 2 Matlab

Realizzata da:
Riccardo Berengan, matricola 2080041
Michele Dioli, matricola 2077629
Gabriele Di Pietro, matricola 2010000

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

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 perchè: 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 dopodichè 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. Applichiamo 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 `l0` 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 Matrice clusterizzata con molteplicità elevata

Effettuiamo 2 piccoli test cambiando solo il valore del punto iniziale del metodo di newton.

```
%% Test 1 alta molteplicità
lambda = [1,1,1,4,1,1,5,1,1,4,5,4];
l0 = 3.9;
toll = 1e-5; %il metodo converge

%% Test 2 bassa molteplicità
lambda = [1,1,1,4,1,1,5,1,1,4,5,4];
l0 = 3.9;
toll = 1e-5; %il metodo converge
```

3.1.2 Test 2 Matrice clusterizzata con valori molto vicini

```
%% Test 1 clusterizzata con valori molto vicini
lambda =[1, 1.01, 1.02, 5, 5.01, 5.02, 10, 10.001, 10.002, 10.003];
l0 = 10.1;
toll = 1e-9; %il metodo converge

%% Test 2 con la stessa matrice cambio solo il punto iniziale
l0= 1.0125;
%In questo caso il metodo di Newton non converge
```

3.1.3 Test 3 Matrice con molteplicità bassa (1-2)

```
%% Prendiamo una matrice di esempio: lambda = [1,1,1,4,5,6];
% Ad ogni iterazione del test proveremo ad aumentare la molteplicità
% aggiungendo un autovalore (ad esempio 4) a lambda
%Definiamo
l0 = 3.9;
toll = 1e-5;
%% Test 1 molteplicità 1
lambda = [1,1,1,4,5,6]; %il metodo converge per l0 = 3.9
%% Proviamo quindi ad diminuire l0 portandolo a 3.5
l0 = 3.5; %in questo caso la molteplicità algebrica è uguale a 0
            % e il metodo non converge;

%% Test 2 molteplicità 2
lambda = [1,1,1,4,5,6,4]; %il metodo converge per l0 = 3.9
%% Proviamo quindi ad diminuire l0 portandolo a 3.5
l0 = 3.5; %in questo caso la molteplicità algebrica è uguale a 1
            % e il metodo non converge;
```

3.1.4 Test 4 Matrice con molteplicità alta (3-7)

```
%% Effettuiamo un test simile al precedente prendendo come base
lambda = [1,1,1,4,5,6,4,4];
toll = 1e-5;
%% Test 1 molteplicità 3
l0 = 3.5; % Il metodo converge ma la molteplicità algebrica è 2
% Riduciamo la tolleranza toll = 1e-9 e l0 = 3.7
l0 = 3.7;
toll = 1e-9; %Il metodo converge e la molteplicità algebrica è 3
```

```
%% Test 2 molteplicità 4
% resettiamo i valori di base e aggiungiamo un autovalore (4)
% alla fine di lambda
lambda = [1,1,1,4,5,6,4,4,4];
l0 = 3.5; % Il metodo converge ma la molteplicità algebrica è 3
            % da qui quindi senza modificare la tollerenza
% Aumentiamo il punto di partenza
l0 = 3.7; % Il metodo converge e la molteplicità algebrica è 4

%% Test 3 molteplicità 5
% aggiungiamo un autovalore (4) alla fine di lambda e partiamo
% da l0 = 3.7
lambda = [1,1,1,4,5,6,4,4,4,4];
l0 = 3.7; % Il metodo converge e la molteplicità algebrica è 5
% Abbassiamo il punto di partenza a 3.5
l0 = 3.5; % Il metodo converge e la molteplicità trovata è giusta
% Abbassiamo il punto di partenza a 3.3
l0 = 3.3; % Il metodo converge ma la molteplicità trovata è 4

%% Test 4 molteplicità 6
% aggiungiamo un autovalore (4) alla fine di lambda
lambda = [1,1,1,4,5,6,4,4,4,4,4];
l0 = 3.7; % Il metodo converge e la molteplicità algebrica è 6
% Abbassiamo il punto di partenza a 3.6
l0 = 3.6; % Il metodo converge e la molteplicità algebrica è 6
% Abbassiamo il punto di partenza a 3.53
l0 = 3.51; % Anche in questo caso converge e la molteplicità algebrica è 6
% Riportiamo l0 a 3.5
l0 = 3.5; % Il metodo converge e la molteplicità trovata è giusta
% Abbassiamo il punto di partenza a 3.4
l0 = 3.4; % Il metodo converge ma la molteplicità trovata è 5

%% Test 5 molteplicità 7
% aggiungiamo un autovalore (4) alla fine di lambda
lambda = [1,1,1,4,5,6,4,4,4,4,4,4];
l0 = 3.5; % Il metodo converge e la molteplicità algebrica è 7
```

4 Conclusioni

Possiamo concludere che il software prodotto gestisce bene il problema e può trovare autovalori anche con stime iniziali lontani, infatti il metodo di Newton converge anche con stime iniziali molto distanti (*ad esempio*, `l0`$= 3.5$ per $\lambda = 4$).
Il software è stato testato con matrici di varie dimensioni e con molteplicità algebriche e geometriche diverse, e in tutti i casi il metodo di Newton converge, a meno che non si scelga un punto iniziale molto distante dall'autovalore da trovare