LABORATORIO DI CALCOLO NUMERICO

Autovalori di matrici: II

Università di Verona

Dott. S. De Marchi

`Verona, 31 gennaio 2005`

Data una matrice quadrata A $n \times n$, a coefficienti reali, i cui autovalori possono essere ordinati come segue:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n| \,.$$

Abbiamo visto la scorsa volta come con il *metodo delle potenze con shift* sia possibile cercare l' autovalore di A, più vicino ad numero η fissato. In pratica si trattava di applicare il *metodo delle potenze inverse* per il calcolo dell'autovalore minimo $\lambda_{min}(A_\eta)$ della matrice $A_\eta = A - \eta\, I$. L'autovalore cercato, dipendente da η, è $\lambda_\eta = \lambda_{min}(A_\eta) + \eta$.

Per individuare un valore η da cui partire, si possono costruire i *cerchi di Gerschgorin*, $C_i^{(r)}$ e $C_i^{(r)}$, $i = 1, ..., n$, associati alle righe e alle colonne di A.

$$C_i^{(r)} = \{z \in \mathbb{C} : \; |z - a_{i,i}| \leq \sum_{j=1, j\neq i}^{n} |a_{i,j}| \; \} \,,$$

$$C_i^{(c)} = \{z \in \mathbb{C} : \; |z - a_{i,i}| \leq \sum_{j=1, j\neq i}^{n} |a_{j,i}| \; \} \,,$$

Usare l'allegato codice `CerchiGerschgorin.m` per costruire i cerchi di Gerschgorin per la matrice dipendente dal parametro α dell'esercitazione del 24.1.05. Quindi individuare gli opportuni shift per calcolarne tutti gli autovalori.

◇◇

Se si desiderano **tutti gli autovalori** di una matrice, bisogna ricorrere a tecniche che consentono dapprima di ridurre la matrice ad una forma più semplice mediante trasformazioni per similitudine pervenendo a una forma triangolare superiore o diagonale: il calcolo degli autovalori diventa così notevolemente semplificato. Questa è la filosofia delle *trasformazioni* con matrici ortogonali di Householder o Givens. Su tale filosofia si basa infatti il *metodo QR e le sue varianti* per autovalori.

Il metodo QR funziona come segue. Sia $A \in \mathbb{R}^{n\times n}$; data $Q^{(0)} \in \mathbb{R}^{n\times n}$ ortogonale (cioè $Q^TQ = I$) e posto $T^{(0)} = (Q^{(0)})^T AQ^{(0)}$, per $k = 1, 2, ...$ finché converge fare i seguenti passi

- mediante la fattorizzazione QR di A (usare `qr`), determinare $T^{(k-1)} = Q^{(k)}R^{(k)}$;
- quindi porre $T^{(k)} = R^{(k)}Q^{(k)}$.

Se A ha autovalori *reali e distinti in modulo* il metodo converge ad una matrice triangolare superiore i cui autovalori stanno sulla diagonale principale. Poiché il metodo ha lo scopo di annullare gli elementi della parte triangolare inferiore sotto la diagonale principale partendo dall'elemento in basso a sinistra, un possibile *test* di arresto è quello di verificare che gli tutti gli elementi della sottodiagonale siano in modulo minori di una prescelta tolleranza ϵ (vedasi l'esercizio 1.)

Se A ha qualche autovalore uguale in modulo, il metodo converge verso una matrice *quasi-triangolare superiore a blocchi* detta anche *decomposizione reale di Schur* di A. Risolvendo poi il problema degli autovalori sui blocchi, si avranno tutti gli autovalori di A (vedasi l'esercizio 2). In tal caso conviene applicare la cosidetta *tecnica dello shift.* Il *metodo QR con shift* consiste nella seguente iterazione: dato lo scalare $\mu \in \mathbb{R}$,

- $T^{(0)} = (Q^{(0)})^T A Q^{(0)}$, matrice in forma di Hessenberg superiore.
- Quindi, per $k = 1, 2, ...$ mediante la fattorizzazione QR di A (usare `qr`), determinare $T^{(k-1)} - \mu I = Q^{(k)} R^{(k)}$;
- quindi porre $T^{(k)} = R^{(k)} Q^{(k)} + \mu I$.

Dalla teoria sul metodo QR con shift, sappiamo che se μ viene scelto cosicchè

$$|\lambda_n - \mu| < |\lambda_i - \mu| \ , \quad i = 1, \ldots, n-1$$

allora l'elemento $t^{(k)}_{n,n-1}$ generato dalla precedente iterazione tende rapidamente a zero al crescere di k. Al limite se μ fosse un autovalore della matrice $T^{(k)}$, e ovviamente anche di A, $t^{(k)}_{n,n-1} = 0$ e $t^{(k)}_{n,n} = \mu$. Questo suggerisce di scegliere

$$\mu = t^{(k)}_{n,n} \, .$$

Con questa scelta la convergenza del metodo è quadratica.

Di questo fatto possiamo tenere conto durante l'esecuzione del metodo QR con shift, controllando il valore di $|t^{(k)}_{n,n-1}|$ e ponendolo uguale a zero se risulta

$$|t^{(k)}_{n,n-1}| < \epsilon \left(|t^{(k)}_{n-1,n-1}| + |t^{(k)}_{n,n}| \right) \, . \tag{1}$$

Questo sarà il test da usare nell'implementazione del metodo QR con shift. Se, la matrice A è in forma di Hessenberg superiore, l'azzeramento di $a^{(k)}_{n,n-1}$ implica che $t^{(k)}_{n,n}$ sarà una buona approssimazione di λ_n. Quindi, noto λ_n la ricerca dei rimanenti autovalori si farà sulla matrice $T^{(k)}(1:n-1, 1:n-1)$, riducendo di volta in volta la dimensione del problema fino a determinare **tutti** gli autovalori di A. In pratica una strategia di deflazione.

Riassumendo, il *metodo QR con shift* si può implementare come segue:

```
[Q,R]=qr(A); T=Q*R; iter=1;
for k=n:-1:2,
    I=eye(k);
    while convergenzaQRShift(T,tol,k)==0 & iter <= iter_max,
     mu=T(k,k);
     [Q,R]=qr(T(1:k,1:k)-mu*I);
     T(1:k,1:k)=R*Q+mu*I;
     iter=iter+1;
    end
T(k,k-1)=0;
end
```

dove il test di convergenza, si farà implementando una funzione Matlab chiamata `convergenzaQRShift` per la disuguaglianza (1).

◇◇

1. Calcolare con il metodo QR su descritto tutti gli autovalori di `A=[30 1 2 3; 4 15 -4 -2; -1 0 3 5; -3 5 0 -1];`. Determinare anche il numero di iterazioni fatte.

2. Si consideri ora la matrice A (tratta dal testo di Quarteroni, Sacco e Saleri *Matematica Numerica*, p. 178)

$$A = \begin{pmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{pmatrix},$$

i cui autovalori (arrotondati a due decimali) sono $\lambda_1 = 65$, $\lambda_{2,3} = \pm 21.28$ e $\lambda_{4,5} = \pm 13.13$. Calcolare tutti gli autovalori sia con il metodo QR che con il metodo QR con shift. Osservare la velocità di convergenza che con la tecnica dello shift è notevolmente accelerata.

◊◊

Tempo massimo: 2 ore.

```
%---------------- CerchiGerschgorin.m---------------
clear
%---------------------------------------------------
% Costruiamo i cerchi di Gerschgorin di una matrice A
%---------------------------------------------------
tol=1.e-10; alfa=input('Dammi alfa = ')
%A=[alfa 2 3 10; 5 12 10 7; 9 7 6 13; 4 16 18 0];
% -> Matrice testo Quarteroni e Saleri, pag. 149 <--
A=[alfa 2 3 13; 5 11 10 8; 9 7 6 12; 4 14 15 1];
%------------------------------------------------
% -> Matrice testo Quarteroni e Saleri, pag. 155 <--
% A=[30 1 2 3; 4 15 -4 -2; -1 0 3 5; -3 5 0 -1];
%------------------------------------------------
Amod=abs(A); n=max(size(A)); raggi=sum(Amod,2)-diag(Amod);
xc=real(diag(A)); yc=imag(diag(A));

% angoli per il disegno dei cerchi
th=[0:pi/100:2*pi];
x=[]; y=[];
 figure(1); clf; axis equal; hold on;

for i=1:n,
    x=[x; raggi(i)*cos(th)+xc(i)];
    y=[y; raggi(i)*sin(th)+yc(i)];
    patch(x(i,:),y(i,:),'red');
end
% disegno il bordo e il centro dei cerchi
for i=1:n,
    plot(x(i,:),y(i,:),'k',xc(i),yc(i),'ok');
end

xmax=max(max(x));ymax=max(max(y));
xmin=min(min(x));ymin=min(min(y)); hold off; figure(2);
 clf; axis equal; hold on;
%-------------------------------------------------------------------
% I cerchi lungo le colonne... sono quelli della matrice trasposta
%--------------------------------------------------------------------
raggi=sum(Amod)-(diag(Amod))'; x=[]; y=[];
for i=1:n,
    x=[x; raggi(i)*cos(th)+xc(i)];
    y=[y; raggi(i)*sin(th)+yc(i)];
    patch(x(i,:),y(i,:),'green');
```

```
end
% disegno il bordo e il centro dei cerchi
for i=1:n,
    plot(x(i,:),y(i,:),'k',xc(i),yc(i),'ok');
end
%Determino il bounding box per il plot ottimale
xmax=max(max(max(x)),xmax); ymax=max(max(max(y)),ymax);
xmin=min(min(min(x)),xmin); ymin=min(min(min(y)),ymin);

hold off; axis([xmin xmax ymin ymax]); figure(1); axis([xmin xmax
ymin ymax]);
```