Tutoraggio #2

Federico Pichi

3 aprile 2019

Esercizio 1

Si definiscano in Matlab le funzioni $f(x) = (1-x)^5$ e $g(x) = 1 - 5x + 10x^2 - 10x^3 + 5x^4 - x^5$ e si noti che in aritmetica esatta $f(x) = g(x)$, $\forall x \in \mathbb{R}$. Si visualizzino sullo stesso grafico le due funzioni f e g nell'intervallo $I_1 = [0.998, 1.002]$ e nell'intervallo $I_2 = [1.998, 2.002]$. Si commentino i risultati ottenuti.

Esercizio 2

Si consideri la matrice di Hilbert $A \in \mathbb{R}^{n\times n}$, tale che $A(i,j) = \frac{1}{i+j-1}$ per $i,j = 1,\dots,n$. Calcolare per $n = 1,\dots,10$ la matrice A (**hilb**) ed il suo numero di condizionamento $K_2(A)$. Salvare i numeri di condizionamento calcolati in un vettore. Visualizzare su un grafico in scala semilogaritmica sulle ordinate l'andamento di $K_2(A)$ in funzione di n (**semilogy**) e commentare il risultato. Si costruisca un vettore colonna x_e unitario di n componenti e si imponga il termine noto $b = A * x_e$, e per ogni $n = 1, 2, \dots 10$ si risolva il sistema lineare $Ax = b$ mediante il comando \ e si calcoli l'errore relativo $err = ||x - x_e||_2/||x_e||_2$. Si visualizzi tale errore su un grafico in scala semilogaritmica e si deduca che esso ha lo stesso comportamento del numero di condizionamento di A.

Esercizio 3

Utilizzando il comando **diag**, definire in Matlab la matrice $A \in \mathbb{R}^{n\times n}, n = 10$ tale che $A(i,i) = 2$, per $i = 1,\dots,n$, ed $A(i,i+1) = -1$, $A(i+1,i) = -1$, per $i = 1,\dots,n-1$. Calcolare successivamente il determinante (**det**), le norme $||A||_1, ||A||_2, ||A||_\infty$ (**norm**) ed i numeri di condizionamento associati (**cond**).

Esercizio 4

Implementare l'algoritmo di fattorizzazione LU ed i metodi di risoluzione in avanti ed all'indietro per sistemi lineari.

Esercizio 5

Creare una matrice random $A \in \mathbb{R}^{n\times n}, n = 10$ ed applicare la decomposizione LU implementata al passo precedente. Calcolare l'errore commesso rispetto alla matrice iniziale. Utilizzare i metodi di risoluzione in avanti e all'indietro per risolvere il sistema $Ax = b$ tale che $x = \text{ones}(10,1)$. Ripetere la procedura per la matrice di Hilbert $H \in \mathbb{R}^{n\times n}, n = 10$.

Esercizio 6

Si implementi il metodo dei cerchi di Gershgorin e lo si applichi alle matrici:

- $A = [2, -0.5, 0, -0.5; 0, 4, 0, 2; -0.5, 0, 6, 0.5; 0, 0, 1, 9]$,
- $B = [-5, 0, 0.5, 0.5; 0.5, 2, 0.5, 0; 0, 1, 0, 0.5; 0, 0.25, 0.5, 3]$,
- $C = [1, 0, -1; 0, -2 + 2i, 2\sqrt{2}; 0.25, 2, -5.75i]$.

Soluzione 1

Le funzioni f e g corrispondono allo stesso polinomio di grado 5. Tuttavia, la valutazione del polinomio mediante l'espressione data in g presenta molte più cancellazioni numeriche quando il valore della funzione è prossimo a zero: infatti in tal caso il calcolo di g consiste nel sommare quantità la cui somma è prossima a zero. Precisamente, I_1 è un intorno di 1, che è uno zero di g, ed è quindi tale che la somma di tutti gli addendi che costituiscono l'espressione di $g(x)$ sia nulla o comunque molto piccola per $x \in I_1$. Da qui le oscillazioni visibili nel grafico di g nell'intervallo I_1. Quando invece il valore del polinomio è ben distinto da zero, come nel caso dell'intervallo I_2, le espressioni di f e g sono equivalenti anche in precisione macchina.

Codice 1: Soluzione 1

```
f_1 = @(x) (1-x).^5;
g_1 = @(x) 1-5*x+10*x.^2-10*x.^3+5*x.^4-x.^5;
x = 0.998 : 5e-5 : 1.002;
figure;
plot(x, f_1(x), 'r', x, g_1(x), 'b+'); grid;
x = 1.998 : 5e-5 : 2.002;
figure;
plot(x, f_1(x), 'r', x, g_1(x), 'b+'); grid;
```

Soluzione 2

Il grafico ottenuto mostra l'andamento del numero di condizionamento e dell'errore in funzione di n in un grafico in scala semilogaritmica. Si nota che il numero di condizionamento ha un andamento lineare nel grafico. Questo dimostra una dipendenza esponenziale di $K_2(A)$ da n. Si nota inoltre che l'errore relativo ha un comportamento analogo al numero di condizionamento, anche se molto più piccolo. Questo conferma il fatto che, più il condizionamento della matrice è grande, più gli errori di round-off vengono amplificati nella risoluzione del sistema lineare portando a un errore relativo sempre maggiore.

Codice 2: Soluzione 2

```
K_2 = zeros(1,10);
err = zeros(1,10);
nn = 1:10;

for n = 1:10
A = hilb(n);
K_2(n) = cond(A);
x_e = ones(n,1);
b = A*x_e;
x = A\b;
err(n) = norm(x-x_e)/norm(x_e);
end

figure
semilogy(nn, K_2, 'r');
grid;
xlabel('n');
ylabel('K_2(n)');

figure
semilogy(nn, K_2, 'r--o', nn, err, 'b-s');
grid; xlabel('n'); legend('K_2(n)', 'err(n)');
```

Soluzione 3

Codice 3: Soluzione 3

```
n=10;
A = 2*diag(ones(1,n)) - diag(ones(1,n-1), 1) - diag(ones(1,n-1), -1);
d = det(A);
```

```
norm_1 = norm(A,1);
norm_2 = norm(A,2);
norm_inf = norm(A,inf);

K_1 = cond(A,1);
K_2 = cond(A,2);
K_inf = cond(A,inf);
```

Soluzione 4

Codice 4: Soluzione 4

```
A=rand(10);
B=lu2(A);
L=tril(B,-1)+(eye(10));
U=triu(B);
M=L*U;
norma=norm(A-M);
b=sum(A,2);
y=forward(L,b);
x_r=backward(U,y);

H=hilb(10);
b=sum(H,2);
A=lu2(H);
L=tril(A,-1)+eye(10);
U=triu(A);
y=forward(L,b);
x_h=backward(U,y);
```

Soluzione 5

Codice 5: Soluzione 5

```
function [ A ] = lu2( A )
%Questo programma fattorizza una matrice A con LU e la restituisce come somma di matrice
%triangolare superiore U e di una matrice strettamente triangolare inferiore L
[n,m]=size(A);
if n~=m
    error('matrice non quadrata')
end
for k=1:n-1
    if A(k,k)==0
        error('elemento pivoting nullo')
    end
    A(k+1:n,k)=A(k+1:n,k)/A(k,k);
    for i=k+1:n
        for j=k+1:n
            A(i,j)=A(i,j)-A(i,k)*A(k,j);
        end
    end
end

end
```

```
function [ b ] = forward(L,b)
%Questo programma risolve un sistema lineare triangolare inferiore
[n,m]=size(L);
if(n~=m) error('attenzione matrice non quadrata')
end
if(prod(diag(L))==0) error('matrice singolare')
end
b(1)=b(1)/L(1,1);
```

```
for i=2:n
    b(i)=(b(i)-L(i,1:i-1)*b(1:i-1))/L(i,i);
end
end
```

```
function [ b ] = backward(U,b)
%Questo programma risolve un sistema lineare triangolare superiore
[n,m]=size(U);
if(n~=m)
    error('attenzione matrice non quadrata')
end
if(prod(diag(U))==0)
    error("matrice singolare')
end
b(n)=b(n)/U(n,n);
for i=n-1:-1:1
    b(i)=(b(i)-U(i,i+1:n)*b(i+1:n))/U(i,i);
end
end
```

Soluzione 6

Consideriamo la matrice A. Dall'esame dei cerchi riga vediamo che c'è un cerchio isolato di centro $(9,0)$ e raggio 1 che potrà contenere un solo autovalore $\lambda_1 \in (8,10)$ che dovrà essere reale. Dall'esame dei cerchi colonna vediamo che ci sono altri due cerchi isolati di raggio 0.5 e centro $(2,0)$ e $(4,0)$ che corrispondono ad altri due autovalori reali, $\lambda_2 \in (1.5,2.5)$ e $\lambda_3 \in (3.4,4.5)$. Consideriamo ora la matrice B. Dall'analisi dei suoi cerchi riga e colonna deduciamo che c'è un solo cerchio isolato di centro $(-5,0)$ e raggio 0.5. Per come sono distribuiti i cerchi esiste quindi un autovalore reale in $(-5.5,-4.5)$. Gli altri tre cerchi hanno invece intersezione non vuota e quindi i restanti tre autovalori di B potranno essere o tutti reali o uno reale e due complessi coniugati. Per la matrice complessa C possiamo notare subito che un autovalore avrà sicuramente parte immaginaria non nulla, quello corrispondente al cerchio con centro $(0,-5.75i)$ e raggio 2.25.

Codice 8: Soluzione 6

```
A = [2 -0.5 0 -0.5; 0 4 0 2; -0.5 0 6 0.5; 0 0 1 9];
gershdisc(A)
B = [-5 0 0.5 0.5; 0.5 2 0.5 0; 0 1 0 0.5; 0 0.25 0.5 3];
gershdisc(B)
C = [1 0 -1; 0 -2+2i 2*sqrt(2); 0.25 2 -5.75i];
gershdisc(C)
```

```
function gershdisc( A )
%La funzione gershdisc prende in input una matrice A e da in output un
%grafico con i dischi di gershgorin.
n=256;
d=2*pi/n;
t=0:d:2*pi;
m=size(A,1);
r=sum(abs(A),2)-diag(abs(A));
rt=(sum(abs(A),1).')-diag(abs(A));
for i=1:m
plot(real(A(i,i))+r(i)*cos(t),imag(A(i,i))+r(i)*sin(t),'b');
hold on
plot(real(A(i,i))+rt(i)*cos(t),imag(A(i,i))+rt(i)*sin(t),'r');
end
aut=eig(A);
plot(real(aut),imag(aut),'*');
grid
axis equal
end
```