

Esercizio

Si consideri, fissato $n \in \mathbb{N}$, la matrice $A \in \mathcal{M}_n(\mathbb{R})$ generata dal comando $A = \text{magic}(n)$; e il sistema lineare $Ax = b$, dove il termine noto $b \in \mathbb{R}^n$ è scelto in modo tale che la soluzione esatta sia $x = (1, 2, \dots, n)^T \in \mathbb{R}^n$.

- 1 E' possibile calcolare la fattorizzazione $PA = LR$? E' possibile calcolare la fattorizzazione QR ? Quale è più conveniente dal punto di vista computazionale e perchè?
- 2 Si risolva il sistema lineare $Ax = b$ utilizzando le funzioni `gauss2`, `solupper` e `sollower` per $n = 10, 11, 12, \dots, 20$. Si calcoli il determinante di A . Si calcolino, inoltre, il rapporto tra la norma infinito del residuo e la norma infinito del termine noto (residuo normalizzato), l'errore relativo commesso, il numero di condizionamento e una stima di tale numero di condizionamento della matrice al variare di n .
- 3 Calcolare la risoluzione del sistema per $n = 19$, usando la fattorizzazione $A = QR$ ottenuta con la routine di Matlab `qr`. Fornire l'errore commesso e il valore assoluto del determinante.

Punto 1 - Teorico

E' sempre possibile calcolare sia la fattorizzazione $PA = LR$ che $A = QR$ di una matrice. Per una matrice densa la prima fattorizzazione è computazionalmente meno costosa; infatti per esempio per matrici quadrate, il metodo di Gauss con pivoting parziale calcola $PA = LR$ con $\mathcal{O}(\frac{n^3}{3})$ operazioni (floats) mentre l'algoritmo QR richiede $\mathcal{O}(4\frac{n^3}{3})$ operazioni (floats). Se una matrice è sparsa senza una struttura particolare, l'algoritmo QR può essere conveniente (richiede una trasformazione di Givens per ogni element non nullo).

Punto 2 - Pratico

- Per $n = 10, 11, 12, \dots, 20$, si ripete:
- generazione del problema test: $A = \text{magic}(n)$; generazione della soluzione esatta $x = (1, 2, \dots, n)^T \in \mathbb{R}^n$; generazione del termine noto con $b=A*x$;
- fattorizzazione con `gauss2`; controllo sulla non singolarità della matrice; se A è non singolare, risoluzione con `solupper` e `sollower`: si calcola una soluzione w ;
- si calcola il residuo ($r=b-A*w$; residuo normalizzato; errore commesso; condizionamento
- si calcola una stima del condizionamento

Punto 3 - Pratico

- Si fissa $n = 19$;
- generazione del problema test: $A = \text{magic}(n)$; generazione della soluzione esatta $x = (1, 2, \dots, n)^T \in \mathbb{R}^n$; generazione del termine noto con $b = A * x$;
- calcolo della fattorizzazione QR con la routine di Matlab `qr`;
- si calcola una soluzione w ;
- si calcola l'errore commesso;
- si calcola il valore assoluto del determinante di A usando la fattorizzazione.

Soluzione

```
clc
close all
clear all
%% punto 2 dell'esercizio
%%
for n=10:20
    fprintf('n=%g\n',n);
    % generazione del problema test
    A=magic(n);
    x=[1:n]';% sol esatta
    b=A*x;
    %%%%%%%%% fattorizzazione
    [L,R,P,deter]=gauss2(A);%% PA=LR
    fprintf('Il determinante di A vale %g\n',deter);
    %%%% test sulla singularit
    if abs(deter)>eps*norm(A,inf)
        %%% soluzione
        z=sollower(L,b(P));
        w=solupper(R,z);% soluzione calcolata
```

Soluzione (cont.)

```
r=b-A*w;
resnor=norm(r,inf)/norm(b,inf);
fprintf('residuo normalizzato=%g\n',resnor);
errrel=norm(x-w,inf)/norm(x,inf);
fprintf('errrel=%g\n',errrel);
%%%%% calcolo inversa
X=eye(n);
for i=1:n
    z=sollower(L,X(P,i));
    X(:,i)=solupper(R,z);
end
%%% X=A\eye(n);
condA=norm(A,inf)*norm(X,inf);
fprintf('N. cond di A=%g\n',condA);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% stima del condizionamento
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y=sollower(L,r(P));
e=solupper(R,y);
kAinv=norm(e,inf)/norm(r,inf)*norm(A,inf);
```

Soluzione (cont.)

```
fprintf('stima del N. cond di A=%g\n',kAinv);
else
    fprintf('La matrice e'' singolare\n');
end

end

%%%%%%%%%%%%%% terzo punto
fprintf('===== \n');
n=19;
A=magic(n);
x=[1:n]'; % sol esatta
b=A*x;
%%%%%%%%%%
[Q,R]=qr(A);
absdet=abs(prod(diag(R)));
if absdet>eps*norm(A,inf)
    w=solupper(R,Q'*b);
    errrel=norm(x-w,inf)/norm(x,inf);
    fprintf('errrel=%g\n',errrel);
else
    fprintf('La matrice e'' singolare\n');
```

Soluzione (cont.)

```
end  
fprintf('val assoluto del determinante=%g\n',absdet);
```