CN AES - Prima Prova in Itinere 19/4/2023

Durata complessiva: 1h 45'. Punteggio: 32
Le funzioni Matlab per la prova sono disponibili alla pagina WeBeeP del corso, cartella Esami, file **FunzioniMATLAB_esame_2023.zip**

Dati Studente

1

Inserire il numero di MATRICOLA
*

981411

2

Selezionare la PENULTIMA cifra X del numero di matricola (yyyyXy)
*

- ○ 0
- ◉ 1
- ○ 2
- ○ 3
- ○ 4
- ○ 5
- ○ 6
- ○ 7
- ○ 8
- ○ 9

Test - 15 punti

Gli studenti aventi diritto a svolgere la prova ridotta del 30% secondo la L.170/2010 (indicazioni **Multichance** team) **NON** svolgono i quesiti contrassegnati con **(***)**

3

Test 1
(1 punto)

1 — 1 pt

Per l'insieme dei numeri floating point $\mathbb{F}(2, 5, L, U)$ dipendente da due parametri $L, U \in \mathbb{Z}$ tali che $L < -2$ e $U > 6$, dati la mantissa $m = (10101)_2$, il segno $s = 0$ e l'esponente $e = 3$, si riporti il numero reale x così rappresentato in base 10.

5.2500

4

Test 2 - (***) No Multichance
(1 punto)

2 — 1 pt (***) No Multichance

Il metodo di Heron consente di approssimare $\sqrt{5}$ applicando il seguente algoritmo: dato x_0, porre $x_{n+1} = \dfrac{x_n + 5/x_n}{2}$ per $n = 0, 1, 2 \ldots$; si ottiene $\lim_{n \to +\infty} x_n = \sqrt{5}$. Quante operazioni elementari vengono effettuate per ottenere x_{50}?

150

5

Test 3
(1 punto)

3 — 1 pt

Si considerino 10 sistemi lineari $A\mathbf{x}_j = \mathbf{b}_j$ per $j = 1, \ldots, 10$, dove la matrice $A \in \mathbb{R}^{90 \times 90}$ è fissata, tridiagonale e a dominanza diagonale stretta per righe, mentre i vettori $\mathbf{b}_j \in \mathbb{R}^{90}$ rappresentano diversi termini noti. Qual è il numero di operazioni richiesto per la risoluzione di tali sistemi lineari per $j = 1, \ldots, 10$ attraverso l'uso computazionalmente più efficiente di un metodo diretto?

4727

6

Test 4 - (***) No Multichance
(2 punti)

4 — 2 pt (***) No Multichance

Si consideri un sistema sovradeterminato $A\mathbf{x} = \mathbf{b}$, dove $A \in \mathbb{R}^{3\times 2}$ ha rango pieno e $\mathbf{b} = (1, 2, 2)^T \in \mathbb{R}^3$. La fattorizzazione QR ridotta di A è tale che $Q = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & 1 \\ \frac{1}{\sqrt{2}} & 0 \end{bmatrix}$ e $R = \begin{bmatrix} 2 & -\gamma \\ 0 & 1 \end{bmatrix}$, dove $\gamma \in \mathbb{R}$ è un parametro positivo ($\gamma > 0$).
Si riporti la soluzione $\mathbf{x}^* \in \mathbb{R}^2$ del sistema nel senso dei minimi quadrati.

(gamma + (3*2^(1/2))/4, 2)'

7

Test 5
(2 punti)

5 — 2 pt

Si consideri la matrice $A = \begin{bmatrix} 4 & -1 & 0 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 \\ 0 & -1 & 3 & 0 & 0 \\ 0 & 0 & 1 & 6 & 0 \\ 0 & 0 & 0 & 9 & -1 \end{bmatrix}$. Per quali valori dello shift $s \in \mathbb{R}$ è possibile applicare il metodo delle potenze inverse con shift per l'approssimazione dell'autovalore 3 di A?

1 < s < 4, s≠3

8

Test 6
(1 punto)

6 — 1 pt

Si consideri la funzione $f(x) = x - \sqrt{5}$, dotata di uno zero $\alpha \in [0, 5]$. Quante iterazioni $k_{min} > 0$ sono necessarie al metodo di bisezione al fine di garantire un errore inferiore a 10^{-3}?

12

9

Test 7
(2 punti)

7 — 2 pt

Si consideri la funzione $\Phi(x) = \frac{x^4}{4} + \frac{x^2}{2} - 3x + 5$. Si applichi il metodo di Newton all'approssimazione del punto di minimo α di Φ partendo dall'iterata iniziale $x^{(0)} = 0$. Si riportino i valori delle iterate $x^{(1)}$, $x^{(2)}$ e $x^{(5)}$ così ottenute con almeno quattro cifre decimali.

x1 = 3,   x2 = 2.0357,   x5 = 1.2144

10

Test 8 - (***) No Multichance
(2 punti)

8 — 2 pt   (***) No Multichance

Si consideri il metodo di Newton modificato per l'approssimazione dello zero $\alpha = 3$ della funzione $f(x) = \sin\left(\frac{\pi}{3}x\right)(x-3)^2$. Si applichi opportunamente tale metodo partendo dall'iterata iniziale $x^{(0)} = 4$ e si riportino i valori delle iterate $x^{(1)}$ e $x^{(2)}$ così ottenute.

x1 = 2.84819157 ,   x2 = 3.00042821

11

Test 9
(1 punto)

9 — 1 pt

Si consideri la funzione di iterazione $\phi(x) = x - \frac{1}{3}\left(1 - e^{1-3x}\right)$ e il suo punto fisso $\alpha = \frac{1}{3}$. Qual è l'ordine di convergenza atteso dal metodo delle iterazioni di punto fisso ad α per $x^{(0)}$ "sufficientemente" vicino ad α?

2

12

Test 10
(2 punti)

10 — 2 pt

Si consideri la funzione di iterazione $\phi(x) = \gamma\left(x^2 - 4x - 5\right) + 5$, dipendente dal parametro $\gamma \in \mathbb{R}$. Relativamente al suo punto fisso $\alpha = 5$, per quali valori di γ il metodo delle iterazioni di punto fisso converge ad α per ogni $x^{(0)}$ "sufficientemente" vicino ad α e in modo da garantire che $\left|x^{(k)} - \alpha\right| < \left|x^{(k+1)} - x^{(k)}\right|$ per ogni $k \geq 0, 1, \ldots$?

-1/6 < gamma < 1/6

Esercizio - 17 punti

Gli studenti aventi diritto a svolgere la prova ridotta del 30% secondo la L.170/2010 (indicazioni **Multichance** team) **NON** svolgono i quesiti contrassegnati con **(***)**

Si consideri il sistema lineare $A\mathbf{x} = \mathbf{b}$, dove $A \in \mathbb{R}^{n\times n}$ è una matrice simmetrica e definita positiva, e $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$ per $n \geq 1$. In particolare, si pongano: $n = 225$, $A \in \mathbb{R}^{225\times 225}$ assegnata con il comando Matlab® seguente

```
>> A = full( gallery( 'poisson', 15 ) )
```

e $\mathbf{b} = (2, 2, \ldots, 2)^T \in \mathbb{R}^{225}$.

13

Punto 1)
(2 punti)

Punto 1) — 2 pt

Si verifichi che la matrice A è simmetrica e definita positiva giustificando tutti i passaggi. Quale metodo diretto è computazionalmente conveniente applicare per risolvere il sistema lineare $A\mathbf{x} = \mathbf{b}$ assegnato? Si giustifichi dettagliatamente la risposta e si riporti in numero di operazioni stimato.

```
n = 225;
A = full(gallery('poisson', 15));
b = 2*ones(n,1);
% verifico che A'== A e che l'autovalore reale più piccolo sia positivo:
if isequal(A,A') && min(eig(A))>0
    disp('A è simmetrica e definita positiva');
else
    disp('A non è simmetrica o definita positiva');
end
%{
Il metodo computazionalmente più conveniente è la fattorizzazione di
Cholesky, in quanto A si è verificato che è simmetrica e definita
positiva e questa fattorizzazione ha un costo computazionale di O((n^3)/3),
la metà della fattorizzazione LU che ha un costo di O((2n^3)/3). Il costo
dell'algoritmo è quindi di (n^3)/3 per la fattorizzazione, di n^2 per
risolvere il sistema triangolare inferiore R'*y=b con sostituzioni in avanti,
e n^2 operazioni per risolvere il sistema triangolare inferiore
R*x=y con sostituzioni all'indietro.
Complessivamente si ha (n^3)/3 + 2*n^2, cioè O(n^3)/3)
%}
```

14

Punto 2)

(3 punti)

Punto 2) — 3 pt

Si applichi tramite Matlab® il metodo diretto di cui al Punto 1) alla soluzione del sistema lineare $A\mathbf{x} = \mathbf{b}$ riportando tutti i passaggi svolti. Dopo aver ottenuto la soluzione numerica $\widehat{\mathbf{x}} \in \mathbb{R}^n$, si stimi l'errore relativo ottenuto $e_{rel} = \dfrac{\|\mathbf{x} - \widehat{\mathbf{x}}\|}{\|\mathbf{x}\|}$. Si definisca tutta la notazione usata e si riportino tutti i comandi Matlab® utilizzati.

```
R = chol(A); % trovo la matrice R tale che R'*R = A
y = fwsub(R',b); % risolvo il sist triangolare inferiore
sol = bksub(R,y);% risolvo il sist triangolare superiore
% controllo che il condizionamento della matrice A non sia troppo alto
cond(A);
% utilizzo la stima dell'errore con il condizionamento in norma 2 e il residuo normalizzato
% in norma 2
err_stim_chol = cond(A)*norm(b-A*sol)/norm(b) %9.0107e-13
```

15

Punto 3)
(2 punti)

Punto 3) — 2 pt

Si consideri ora il metodo di *Jacobi* per la soluzione del sistema lineare $A\mathbf{x} = \mathbf{b}$. Si verifichi che il metodo di Jacobi converge per ogni scelta dell'iterata iniziale $\mathbf{x}^{(0)} \in \mathbb{R}^n$ e, senza applicare il metodo, si *stimi* il fattore di abbattimento dell'errore $\dfrac{\|\mathbf{x}^{(k)} - \mathbf{x}\|}{\|\mathbf{x}^{(0)} - \mathbf{x}\|}$ dopo $k = 100$ iterazioni del metodo. Si motivi dettagliatamente la risposta, definendo la notazione e riportando i comandi Matlab® usati.

```
% calcolo il raggio spettrale della matrice di iterazione
P = diag(diag(A));
B = eye(n)-P\A;
rho_jac = max(abs(eig(B))) % 0.9808
% vedo il metodo di Jacobi come un caso generale di richardson stazionario
% con alpha ottimale = 1, infatti:
a_opt = 2/(max(eig(P\A))+min(eig(P\A))); % = 1
K_jac = max(eig(P\A))/min(eig(P\A));
d_jac = (K_jac-1)/(K_jac+1);
k_jac = 100;
abb = d_jac^k_jac   % 0.1437
```

16

Punto 4) - (***) No Multichance
(2 punti)

Punto 4) — 2 pt (***) No Multichance

Si consideri ora il metodo del *gradiente* per la soluzione del sistema lineare $A\mathbf{x} = \mathbf{b}$ a cui è associata la funzione energia $\Phi : \mathbb{R}^n \to \mathbb{R}$, dove $\Phi(\mathbf{y}) := \frac{1}{2}\mathbf{y}^T A\mathbf{y} - \mathbf{y}^T\mathbf{b}$. Scelta l'iterata inziale $\mathbf{x}^{(0)} = \mathbf{b}$, si calcolino e si riportino i valori $\Phi\left(\mathbf{x}^{(0)}\right)$ e $\Phi\left(\mathbf{x}^{(1)}\right)$, essendo $\mathbf{x}^{(1)}$ l'iterata ottenuta applicando un'iterazione del metodo del gradiente.

```
phi = @(y) 0.5*y'*A*y - y'*b;
x0 = b;
% applico un'itera del metodo del gradiente ( richardson.m senza
% precondizionamento e non dichiarando alpha, --> richardson dinamico --> gradiente)
x1 = richardson(A,b,eye(n),x0,1e-10,1);
phi(x0) % -780
phi(x1) % -1.6603e+03
```

17

Punto 5 - (***) No Multichance
(3 punti)

Punto 5) — 3 pt (***) No Multichance

Si consideri ora il metodo del *gradiente precondizionato* con matrici di precondizionamento

$$P_1 = \text{tridiag}\,(-1,\, 4,\, -1) \in \mathbb{R}^{n\times n}$$

e $P_2 \in \mathbb{R}^{n\times n}$ ottenuta tramite i seguenti comandi Matlab® :

```
>> R2 = ichol( sparse( A ) ); P2 = R2' * R2;
```

Senza applicare esplicitamente il metodo, si determini per quale delle due matrici di precondizionamento il metodo del gradiente precondizionato converge più rapidamente a $\mathbf{x}$ per ogni scelta di $\mathbf{x}^{(0)}$.

Per la matrice di precondizionamento per cui il metodo converge più rapidamente, si *stimi* il fattore di abbattimento dell'errore $\dfrac{\|\mathbf{x}^{(k)} - \mathbf{x}\|_A}{\|\mathbf{x}^{(0)} - \mathbf{x}\|_A}$ dopo $k = 20$ iterazioni del metodo. Si motivi dettagliatamente la risposta, definendo la notazione e riportando i comandi Matlab® usati.

```
P1 = diagonals([-1 4 -1],n);
R2 = ichol(sparse(A));
```

```
P2 = R2'*R2;

K1 = max(eig(P1\A))/min(eig(P1\A)) % 58.2413
K2 = max(eig(P2\A))/min(eig(P2\A)) % 10.0966
% P2 ha un condizionamento spettrale più basso,
% converge più velocemente poiché a parità di fattore
% di abbattimento impiega meno iterazioni
c = (sqrt(K2)-1)/(sqrt(2)+1);
k = 20;
% |xk-x|_A <= (2*c^k)/(1+c^(2*k))*|x0-x|_A
% abb = (2*c^k)/(1+c^(2*k))
abb = (2*c^k)/(1+c^(2*k)) % 0.2499

%% DEFINIZIONE FUNZIONE

function M = diagonals(v,n)
   u = length(v);
   M = zeros(n);
   if mod(u,2)~=0
      iter = ceil(u/2);
      ind1 = iter;
      ind2 = iter;
      for i = 0 : iter-1
         if i == 0
            M = M + diag(v(ind1)*ones(n-i,1),i);
         else
            M = M + diag(v(ind1)*ones(n-i,1),i);
            M = M + diag(v(ind2)*ones(n-i,1),-i);
         end
         ind1 = ind1 + 1;
         ind2 = ind2 -1;
      end
   else
      iter = ceil(u/2);
      v = [v(1:iter) 0 v(iter+1:end)];
      iter = iter + 1;
      ind1 = iter;
      ind2 = iter;
      for i = 0 : iter-1
         M = M + diag(v(ind1)*ones(n-i,1),i);
         M = M + diag(v(ind2)*ones(n-i,1),-i);
         ind1 = ind1 + 1;
         ind2 = ind2 -1;
      end
   end
end
```

18

Punto 6)
(2 punti)

Punto 6) — 2 pt

Si consideri il metodo delle potenze per approssimare l'autovalore $\lambda_1(A)$. Si applichi il metodo, a partire dall'iterata iniziale $\mathbf{x}^{(0)} = \mathbf{1} \in \mathbb{R}^n$, modificando opportunamente la funzione `eigpower.m` per *stimare* l'ordine e il fattore di convergenza asintotico ottenuti applicando tale metodo iterativo. Si riportino i comandi Matlab® usati e si giustifichi il risultato ottenuto.

```
x0 = ones(n,1);
l1 = max(eig(A));
[lambda,x,iter,lambda_vect]=eigpower(A,1e-10,1e5,x0);
% utilizzo come errore il valore assoluto della differenza tra l'autovalore all'iterata k e
%l'autovalore trovato con la funzione eig
err = abs(l1-lambda_vect);
% utilizzo stimap per vedere con che ordine p converge l'errore
[p, c] = stimap(err);
ordine = ceil(p(end)) % = 1
% per calcolare il fattore suppongo di aver fatto molte iterazioni ( in particolare
% iter = 263, quindi il limite
% lim k->inf (e(k+1)/e(k)^p) = (e(iter)/e(iter-1)^p)
fatt = err(end)/(err(end-1)^ordine) % 0.9260 che è circa uguale al fattore di riduzione %
trovato da stimap:
c(end) % 0.92535749
```

19

Punto 7)
(3 punti)

Punto 7) — 3 pt

Data una generica matrice $A \in \mathbb{R}^{n \times n}$ simmetrica e definita positiva, i suoi autovalori $\{\lambda_i(A)\}_{i=1}^n \in \mathbb{R}$ si possono approssimare applicando il seguente algoritmo che migliora l'efficienza del metodo delle iterazioni QR.

Algorithm 1: Metodo delle iterazioni QR con shift

porre $A^{(0)} = A$;
porre $\mu^{(0)} = 0$;
for $k = 0, 1, \ldots,$ *fino a che un criterio d'arresto è soddisfatto* **do**
  determinare la fattorizzazione QR (ridotta) di $A^{(k)}$, ovvero le matrici $Q^{(k+1)}$ e $R^{(k+1)}$ tali che $Q^{(k+1)}\, R^{(k+1)} = A^{(k)} - \mu^{(k)}\, I$;
  porre $A^{(k+1)} = R^{(k+1)}\, Q^{(k+1)} + \mu^{(k)}\, I$;
  porre $\mu^{(k+1)} = \left(A^{(k+1)}\right)_{n,n}$;
  $\lambda_i^{(k+1)} = (A^{(k+1)})_{ii}$ per $i = 1, \ldots, n$;
end

Si implementi il precedente algoritmo modificando per esempio la funzione Matlab® **`qrbasic.m`** e usando laddove necessario il comando Matlab® **`qr`**.

Si applichi l'algoritmo alla matrice A assegnata. Si riportino le approssimazioni $\lambda_2^{(1)}$, $\lambda_2^{(2)}$ e $\lambda_2^{(20)}$ di $\lambda_2(A)$ così ottenute. Si riportino la funzione Matlab® implementata e i comandi Matlab® usati.

```
l_1 = qrbasic(A,1e-8,1);
l_1(2) % 4.5500
l_2 = qrbasic(A,1e-8,2);
l_2(2) % 4.9555
l_20 = qrbasic(A,1e-8,20);
l_20(2) % 7.3791


% per confrontare:
l = sort(eig(A),'descend');
l(2);

%% FUNZIONE
function D=qrbasic(A,tol,nmax)


[n,m]=size(A);
if n ~= m
  error('La matrice deve essere quadrata')
end
T = A;
mu = 0;
niter = 0;
test = max(max(abs(tril(T,-1))));
while niter < nmax && test > tol
  [Q,R]=qr(T-mu*eye(n));
  T = R*Q + mu*eye(n);
  mu = T(n,n);
  niter = niter + 1;
  test = max(max(abs(tril(T,-1))));
end
if niter > nmax
 fprintf(['Il metodo non converge nel massimo',...
            ' numero di iterazioni permesso']);
else
 fprintf('Il metodo converge in %d iterazioni\n',niter)
end
D = diag(T);
return
```

Consegna

Sottomettere solo a prova conclusa. In caso di **ritiro**, avvisare il docente e successivamente sottomettere il quiz Forms. Selezionare invio **conferma** di sottomissione tramite email