Prova d'esame di Laboratorio di Calcolo I
per il corso di laurea in Matematica
22 Giugno 2010

Tema d'esame: Calcolo degli autovalori di una matrice simmetrica, utilizzando in modo combinato *il metodo delle potenze* e *la procedura di ortonormalizzazione alla Gram–Schmidt.*

Descrizione del metodo di calcolo

È ben noto che gli autovettori $\{\underline{v}_j\}_{j=0}^{n-1}$ di una matrice simmetrica A sono ortogonali tra di loro, cioè sussistono le seguenti due equazioni:

$$A\underline{v}_j = \lambda_j \underline{v}_j \qquad \forall\, j = 0\,,\ \ldots\ ,\, n-1 \tag{1}$$

e

$$\underline{v}_i \cdot \underline{v}_j = \delta_{i,j} \qquad \forall\, i\,,\, j = 0\,,\ \ldots\ ,\, n-1\ , \tag{2}$$

dove λ_j è l'autovalore associato a $\underline{v}_j$ e $\delta_{i,j}$ è la ben nota "delta di Kronecker". Dalla (2) si deduce che gli autovettori $\{\underline{v}_j\}_{j=0}^{n-1}$ sono stati scelti di norma uguale a 1.

Il *metodo delle potenze* si applica a delle matrici tali che esiste un autovalore che è strettamente maggiore in valore assoluto di tutti gli altri. In modo più restrittivo, assumeremo che

$$|\lambda_0| > |\lambda_1| > \ \ldots\ > |\lambda_{n-1}|\ . \tag{3}$$

Per poter applicare il *metodo delle potenze*, si pone quindi un primo problema pratico: occorre trovare un modo per determinare delle matrici simmetriche $\mathcal{U}$ non banali i cui autovalori soddisfano la relazione (3). A tale scopo possiamo procedere nel modo seguente: inizialmente definiamo

$$\mathcal{U} = \mathrm{diag}\big\{\lambda_0\,,\, \lambda_1\,,\ \ldots\ ,\, \lambda_{n-1}\big\}\ , \tag{4}$$

cioè $\mathcal{U}$ è una matrice i cui elementi sono tutti uguali a zero, eccetto quelli lungo la diagonale, dove compaiono in sequenza degli autovalori $\{\lambda_j\}_{j=0}^{n-1}$, che soddisfano l'ipotesi (3). A questo punto, eseguiamo ripetutamente le seguenti tre operazioni:

(i) (ri)definiamo la matrice $\mathcal{U}'$ in modo che sia $\mathcal{U}' = \mathcal{U}$;

(ii) scegliamo i, j e $\vartheta_{i,j}$, in modo tale che (i,j) è una fissata coppia di indici compresi tra 0 e $n-1$, $\vartheta_{i,j}$ è un angolo che dipende dalla suddetta coppia di indici;

(iii) ridefiniamo $\mathcal{U}$ calcolando il prodotto matriciale

$$\mathcal{U} = \mathcal{R}_{i,j}(\vartheta_{i,j}) \cdot \mathcal{U}' \cdot \big(\mathcal{R}_{i,j}(\vartheta_{i,j})\big)^T\ ,$$

dove $\mathcal{R}_{i,j}(\vartheta_{i,j})$ è una matrice ortogonale tale che:

$$\mathcal{R}_{i,j}(\vartheta_{i,j}) = \begin{pmatrix} r_{0,0} & \ldots & r_{0,n-1} \\ \vdots & \ldots & \vdots \\ r_{n-1,0} & \ldots & r_{n-1,n-1} \end{pmatrix} \quad \text{con} \quad r_{l,m} = \begin{cases} \delta_{l,m} & \text{se } (l,m) \notin \big\{(i,i)\,,\,(i,j)\,,\,(j,i)\,,\,(j,j)\big\} \\ \cos\vartheta_{i,j} & \text{se } (l,m) = (i,i) \\ -\sin\vartheta_{i,j} & \text{se } (l,m) = (i,j) \\ \sin\vartheta_{i,j} & \text{se } (l,m) = (j,i) \\ \cos\vartheta_{i,j} & \text{se } (l,m) = (j,j) \end{cases}\ ,$$

cioè $\mathcal{R}_{i,j}(\vartheta_{i,j})$ è una matrice di rotazione con angolo $\vartheta_{i,j}$ sul piano individuato dalle i–esima e j–esima dimensione, mentre è semplicemente l'identità rispetto a tutte le altre coordinate.

Dopo aver iterato ripetutamente le operazioni (i)–(iii) precedentemente descritte, se le scelte di i , j e $\vartheta_{i,j}$ non sono state volutamente banali, alla fine otterremo una matrice $\mathcal{U}$ che è *simmetrica, non diagonale e i suoi autovalori* $\{\lambda_j\}_{j=0}^{n-1}$ *soddisfano l'ipotesi* (3).

Ricordiamo che noi stiamo considerando un problema in cui supponiamo solo di sapere che è soddisfatta l'ipotesi (3), senza conoscere esattamente gli autovalori $\{\lambda_j\}_{j=0}^{n-1}$. Possiamo quindi applicare una prima volta il *metodo delle potenze*, in modo da determinare λ_0 e il corrispondente autovettore $\underline{v}_0$. Nel seguito di questa introduzione, descriveremo come si può utilizzare il ben noto *algoritmo di ortonormalizzazione di Gram–Schmidt.*

Sia $\underline{v}_0 = (v_{0,0}\,,\,\ldots\,,\,v_{0,n-1})^T$ la decomposizione dell'autovettore normalizzato $\underline{v}_0$ (che si intende come "vettore colonna"), allora definiamo inizialmente la matrice U in modo tale che

$$U = \begin{pmatrix} v_{0,0} & v_{0,1} & \cdots & v_{0,n-1} \\ 0 & & & \\ \vdots & & \mathbf{I} & \\ 0 & & & \end{pmatrix}\,, \tag{5}$$

dove $\mathbf{I}$ rappresenta la matrice "identità" $(n-1)\times(n-1)$–dimensionale. A meno che la scelta delle matrici $\mathcal{R}_{i,j}(\vartheta_{i,j})$ descritte precedentemente al punto (iii) sia stata proprio infelice, in generale avviene che $\det U = v_{0,0} \neq 0$ e quindi possiamo ottenere un'opportuna matrice ortogonale trasformando la U come segue:

(I) a partire dalle righe della matrice U , i vettori $\{\underline{u}_j\}_{j=0}^{n-1}$ siano inizialmente definiti, in modo tale che

$$U = \begin{pmatrix} \underline{u}_0^T \\ \vdots \\ \underline{u}_{n-1}^T \end{pmatrix}\;;$$

(II) si iterino le istruzioni descritte ai seguenti punti (III)–(V), mentre l'indice intero i va da 0 a $n-1$;

(III) si iterino le istruzioni descritte ai seguenti punti (IV)–(V), mentre l'indice intero j va da $i+1$ a $n-1$;

(IV) si definisca il vettore temporaneo $\underline{w}$ in modo tale che

$$\underline{w} = \underline{u}_j - \big(\underline{u}_i \cdot \underline{u}_j\big)\underline{u}_i\;;$$

(V) si ridefinisca il vettore $\underline{u}_j$ in modo che sia uguale a $\underline{w}$ normalizzato, cioè

$$\underline{u}_j = \underline{w}/\|\underline{w}\|\;.$$

Alla fine dell'esecuzione della procedura che abbiamo appena descritto, otteniamo che la *nuova* matrice U definita come al punto (I) a partire dai *nuovi* "vettori riga" $\underline{u}_0^T\,,\,\ldots\,,\,\underline{u}_{n-1}^T$ è *ortogonale.* Inoltre, si osserva facilmente che la prima riga non è cambiata rispetto al suo valore iniziale, quindi è ancora $\underline{v}_0^T$, dove ricordiamo che $\underline{v}_0$ è l'autovettore della matrice

A relativo all'autovalore λ_0 . Quando viene effettuato il cambio di base ortogonale relativa ad U , pertanto, la matrice corrispondente ad A è data da

$$U \cdot A \cdot U^T = \begin{pmatrix} \lambda_0 & 0 & \dots & 0 \\ 0 & a'_{1,1} & \dots & a'_{1,n-1} \\ \vdots & \vdots & & \vdots \\ 0 & a'_{n-1,1} & \dots & a'_{n-1,n-1} \end{pmatrix} . \tag{6}$$

Sia $A' = \left(a'_{i,j}\right)_{i,j=1,\,\dots\,,\,n-1}$ la matrice $(n-1)\times(n-1)$–dimensionale che sta nell'angolo a destra in basso della formula precedente. Dall'equazione agli autovalori (1) (che vale per A), dall'ortogonalità di U e dall'ipotesi (3) segue che gli autovalori di A' sono λ_1 , ... , λ_{n-1} e quindi sussiste la seguente catena di disuguaglianze

$$|\lambda_1| > |\lambda_2| > \ \dots\ > |\lambda_{n-1}| \ . \tag{7}$$

Quest'ultima osservazione permette di iterare il nostro metodo. Infatti, se si ridefinisce $A = A'$, dopo aver opportunamente riscalato gli indici e ridefinito $n = n - 1$, possiamo riapplicare tutto il metodo a partire dall'equazione (1) fino alla (7). Se ripetiamo l'algoritmo fino a quando scriviamo l'equazione (6) per matrici bidimensionali, siamo in grado di determinare tutti gli autovalori.

Obiettivo (intermedio) 1:
si scriva un programma in linguaggio **C** che **crea** *una matrice simmetrica non banale con autovalori noti*. Il programma deve contenere:

(A) una *function* che ha cinque argomenti: i valori della dimensione effettiva n , di due indici i e j , di un angolo ϑ e, infine, una matrice $\mathcal{R}$; tale *function* restituisce (attraverso il quinto argomento) la matrice $\mathcal{R}_{i,j}(\vartheta)$ definita come al punto (iii);

(B) una *function* che ha quattro argomenti: due matrici A e B , il valore della dimensione effettiva n e, infine, una terza matrice C ; tale *function* restituisce (attraverso il quarto argomento) il risultato del prodotto matriciale $C = A \cdot B$;

(C) la *main function* che deve essere strutturata in modo tale che, a sua volta, essa contenga:

- (C1) l'*input* della dimensione effettiva n , il cui valore deve essere compreso tra 2 e $NDIM$;
- (C2) l'*input* degli autovalori λ_0 , ... , λ_{n-1} ;
- (C3) la definizione iniziale della matrice $\mathcal{U}$ in modo che sia diagonale e che rispetti l'equazione (4);
- (C4) due *cicli annidati*: il primo per i che va da 0 a $n-1$ e il secondo per j che va da $i+1$ a $n-1$; all'interno di questi due cicli viene ripetuta l'esecuzione delle istruzioni descritte ai seguenti punti (C41)–(C43);
- (C41) si scelga a piacere un valore di $\vartheta = \vartheta_{i,j}$;
- (C42) si determini la matrice $\mathcal{R}_{i,j}(\vartheta_{i,j})$ effettuando la *chiamata della function descritta al punto (A)*;
- (C43) si ridefinisca prima la matrice $\mathcal{U}'$ in modo che sia $\mathcal{U}' = \mathcal{U}$ e poi la matrice $\mathcal{U}$ grazie ad un paio di opportune *chiamate della function descritta al punto (B)*, in modo da effettuare il prodotto matriciale descritto al punto (iii);

(C5) la *stampa ordinata su di un file di output* del valore della dimensione effettiva n e poi (nelle $n \times n$ righe che seguono la prima) di tutti gli elementi della matrice $\mathcal{U}$ (definita come descritto ai precedenti punti (C3)–(C43)), in modo tale che ciascuna riga sia occupata dal valore di un solo elemento di matrice.

Alcuni consigli

È sicuramente opportuno stabilire il valore *costante* di $NDIM$ per mezzo di una direttiva `#define`.

La scrittura del programma è enormemente facilitata se si ricorre al seguente "trucco": quando si deve decidere quanta parte della memoria deve essere destinata ad ospitare i vari elementi dei vettori e delle matrici, si effettuino dei "sovradimensionamenti" in modo tale da allocare sempre, rispettivamente, $NDIM$ e $NDIM \times NDIM$ celle di memoria. Ciò nonostante, le istruzioni che permettono il calcolo dei valori dei componenti dei vettori e delle matrici verranno effettuate tenendo conto che la *dimensione effettiva* dello spazio vettoriale che stiamo considerando è $n \leq NDIM$.

È sicuramente comodo (e *prudente*) utilizzare delle *functions* o parti di programma, che sono incluse in altri programmi precedentemente scritti dagli studenti stessi o dal docente (e reperibili in rete).

Obiettivo (intermedio) 2:

si scriva un secondo programma in linguaggio **C** che è costituito dalla sola *main function*, all'interno della quale tutti i valori degli $n \times n$ elementi della matrice A *vengono letti proprio dal file*, la cui scrittura era stata richiesta al punto (C5) dell'obiettivo 1. Inoltre, dopo aver terminato la lettura da *file*, si *stampi ordinatamente sul video* la matrice A, in modo tale che $\forall\ i = 0, \ldots, n-1$ tutti e soli gli n elementi $a_{i,0}, \ldots, a_{i,n-1}$ della i–esima riga della matrice A *se ne stiano ben allineati sulla stessa riga dello schermo.*

Obiettivo (intermedio) 3:

si integri il programma richiesto dall'obiettivo 2, in modo tale da calcolare un autovalore (che è il massimo in valore assoluto) e il corrispondente autovettore, utilizzando il *metodo delle potenze inverse.* A tal fine si proceda come segue:

(A) si inserisca una *function* che ha quattro argomenti: una matrice A, un vettore $\underline{v}$, il valore della dimensione effettiva n e un vettore $\underline{w}$; tale *function* restituisce (attraverso il quarto argomento) il vettore risultato del seguente prodotto matrice per vettore $\underline{w} = A\underline{v}$;

(B) si inserisca una *function* che ha tre argomenti: due vettori $\underline{v}$ e $\underline{w}$, inoltre, il valore della dimensione effettiva n; tale *function* restituisce il valore del prodotto scalare $\underline{v} \cdot \underline{w}$;

(C) si scriva una *function* che ha due argomenti: un vettore $\underline{v}$ e il valore della dimensione effettiva n; tale *function* restituisce il valore della norma di $\underline{v}$, cioè $\|v\| = \sqrt{\sum_{j=0}^{n-1} v_j^2}$;

(D) si scriva una *function* che ha tre argomenti: una matrice A, il valore della dimensione effettiva n e un vettore $\underline{v}$; tale *function* restituisce un autovalore e (attraverso il terzo argomento) il corrispondente autovettore; il calcolo deve essere effettuato applicando il *metodo delle potenze*, descritto come ai seguenti punti (D1)–(D3);

(D1) si introducano le seguenti definizioni come (molto) rozze approssimazioni iniziali dell'autovalore (che d'ora innanzi chiamiamo λ) e del corrispondente autovettore: $\lambda = 0$, $\underline{v} = \left(1/\sqrt{n}, \ldots, 1/\sqrt{n}\right)$;

(D2) si iterino le operazioni descritte ai seguenti punti (D21)–(D25) mentre è verificata la condizione (di permanenza nel ciclo), che è espressa al punto (D3);

(D21) si ponga $\overline{\lambda} = \lambda$;

(D22) si (ri)calcolino i valori degli elementi del vettore $\underline{w} = A\underline{v}$, effettuando la *chiamata della function descritta al punto (A)*;

(D23) si ricalcoli l'autovalore, effettuando la *chiamata della function descritta al punto (B)*, in modo da porre $\lambda = \underline{v} \cdot \underline{w}$;

(D24) si introduca il vettore $\Delta\underline{w}$, le cui componenti sono calcolate in modo tale che $\Delta\underline{w} = \underline{w} - \lambda\underline{v}$;

(D25) si ridefinisca il vettore $\underline{v}$ in modo tale che

$$\underline{v} = \frac{\underline{w}}{\|\underline{w}\|} \ ,$$

dove la norma euclidea $\|\underline{w}\|$ viene calcolata effettuando la *chiamata della function descritta al punto (C)*;

(D3) se l'errore relativo riguardante la determinazione dell'autovettore è significativamente maggiore dell'*errore di macchina*, cioè se si verifica che

$$\frac{\|\Delta\underline{w}\|}{\|\underline{w}\|} > 2n \ \times 10^{-16} \ ,$$

allora si torni a ripetere le operazioni descritte ai precedenti punti (D21)–(D25);

(E) all'interno della *main function*, si effettui l'opportuna *chiamata della function descritta al punto (D)* e, di seguito, si visualizzi il valore dell'approssimazione finale di λ , che è stata così ottenuta.

Obiettivo (intermedio) 4:
si integri il programma richiesto dall'obiettivo 3, in modo tale da calcolare la trasformata della matrice A , che è costituita da due blocchi, di cui il primo è costituito dal solo autovalore determinato durante il conseguimento dell'obiettivo 3, mentre il secondo è $(n-1) \times (n-1)$–dimensionale (così come descritto nella formula (6)). A tal fine si proceda come segue:

(A) si scriva una *function* che ha tre argomenti: una matrice A , il valore della dimensione effettiva n e un autovettore $\underline{v}$ della stessa matrice A (il quale è stato calcolato così come descritto nell'obiettivo 3); tale *function* restituisce (attraverso il primo argomento) la trasformata della matrice A , cioè $U \cdot A \cdot U^T$; il calcolo deve essere effettuato applicando l'*algoritmo di ortonormalizzazione di Gram–Schmidt*, così come descritto ai seguenti punti (A1)–(A3);

(A1) si definisca una matrice U , in modo tale che sia soddisfatta l'equazione (5);

(A2) si trasformi la matrice U in modo che divenga ortogonale, applicando il procedimento di *ortonormalizzazione alla Gram–Schmidt*, riassunto dalle prescrizioni (I)–(V), che sono riportate nella "descrizione del metodo di calcolo"; si noti che, quando questo algoritmo viene tradotto nel linguaggio di programmazione **C**, *non c'è alcun bisogno di allocare la memoria* per n vettori allo scopo di rappresentare $\{\underline{u}_j\}_{j=0}^{n-1}$;

(A3) si ridefinisca la matrice A in modo che sia uguale al prodotto matriciale $U \cdot A \cdot U^T$;

(B) all'interno della *main function*, si effettui l'opportuna *chiamata della function descritta al punto (A)* e, di seguito, si *stampino ordinatamente sul video* i nuovi valori degli elementi della matrice A (che a questo punto sarà uguale a $U \cdot A \cdot U^T$, dove qui con A s'intende quella definita inizialmente). Tale *stampa ordinata* deve essere come quella già richiesta dall'obiettivo 2, cioè tale che $\forall\ i = 0\,, \ldots\,, n-1$ tutti e soli gli n elementi $a_{i,0}\,, \ldots\,, a_{i,n-1}$ della i–esima riga della matrice A *se ne stiano ben allineati sulla stessa riga dello schermo*.

Un'osservazione banale

Si può considerare che l'obiettivo 4 (e anche i precedenti) sia stato conseguito con successo, se la stampa della (ridefinita) matrice A, così come è richiesta al precedente punto (B), mette in evidenza la stessa struttura a blocchi descritta nel membro di destra dell'equazione (6).

Obiettivo (finale) 5:

si integri il programma richiesto dall'obiettivo 4, in modo tale da determinare tutti gli autovalori della matrice A di partenza, iterando il metodo descritto in precedenza. A tal fine si proceda come segue:

(A) all'interno della *main function*, si ripetano le operazioni descritte ai seguenti punti (A1)–(A4), mentre è verificata la condizione $n > 2$:
  - (A1) si ridefinisca la matrice A in modo che sia uguale al blocco $(n-1) \times (n-1)$–dimensionale, che appare in basso a destra nel membro di destra dell'equazione (6);
  - (A2) si decrementi di 1 il valore di n;
  - (A3) si riesegua quanto richiesto al punto (E) dell'obiettivo (3);
  - (A4) si riesegua quanto richiesto al punto (B) dell'obiettivo (4).