Prova d'esame di Laboratorio di Calcolo I
per il corso di laurea in Matematica
8 Luglio 2013

Tema d'esame: Calcolo degli autovalori e di alcuni opportuni autospazi di una matrice tale che tutti i suoi autovalori sono complessi coniugati a coppie.

Descrizione del metodo di calcolo

Sia la matrice U a coefficienti *reali* tale che i suoi autovalori $\{\lambda_j\}_{j=0}^{n-1}$ soddisfano le seguenti proprietà

$$(1) \qquad \big|\lambda_0\big| < \big|\lambda_2\big| < \ldots < \big|\lambda_{n-2}\big| \ , \qquad \big|\lambda_{2j}\big| = \big|\lambda_{2j+1}\big| \ \ \forall\ j = 0, \ldots, n/2 - 1 \ ,$$

dove d'ora in avanti si intende che la dimensione n deve essere pari. Siccome è ben noto che gli autovalori di una matrice reale sono a sua volta reali o complessi coniugati a coppie, la seconda richiesta espressa in formula (1) implica che esistono $n/2$ angoli $\{\vartheta_j\}_{j=0}^{n/2-1}$, tali che

$$\lambda_{2j} = |\lambda_{2j}| \exp(\mathbf{i}\vartheta_j) \ , \qquad \lambda_{2j+1} = |\lambda_{2j}| \exp(-\mathbf{i}\vartheta_j) \ , \qquad \forall\ j = 0, \ldots, n/2 - 1 \ .$$

Dalla formula (1), segue immediatamente che la matrice

$$(2) \qquad U^T U = \begin{pmatrix} a_{0,0} & \cdots & a_{0,n-1} \\ \vdots & \vdots & \vdots \\ a_{n-1,0} & \cdots & a_{n-1,n-1} \end{pmatrix} \ ,$$

(che è evidentemente simmetrica) ha tutti gli autovalori che sono positivi e di molteplicità 2. In altri termini, sia $\mathcal{P}$ il *polinomio caratteristico* della matrice $U^T U$, cioè

$$(3) \qquad \mathcal{P}(\beta) = \det\big(U^T U - \beta \mathbf{I}\big) \ ,$$

allora le n soluzioni dell'equazioni $\mathcal{P}(\beta) = 0$ sono le seguenti:

$$(4) \qquad |\lambda_0|^2 = |\lambda_1|^2 \ , \quad |\lambda_2|^2 = |\lambda_3|^2 \ , \quad \ldots \quad |\lambda_{n-2}|^2 = |\lambda_{n-1}|^2 \ .$$

Sia

$$(5) \qquad R_i = \sum_{\substack{0 \le j < n \\ j \ne i}} |a_{i,j}| \qquad \forall\ 0 \le i \le n-1 \ ,$$

allora il primo teorema di Gershgorin implica che

$$|\lambda_{2j}|^2 = |\lambda_{2j+1}|^2 \in \bigcup_{i=0}^{n-1} \mathcal{B}_{R_i}(a_{i,i}) \qquad \forall\ j = 0, \ldots, n/2 - 1 \ .$$

La precedente relazione può essere espressa più esplicitamente come segue. Siano gli estremi c e d tali che

$$(6) \qquad c = \min_{0 \le i \le n-1} \big\{ a_{i,i} - R_i \big\} \ , \qquad d = \max_{0 \le i \le n-1} \big\{ a_{i,i} + R_i \big\} \ ,$$

allora sussiste la seguente catena di disuguaglianze

$$c \leq \left|\lambda_0\right|^2 < \left|\lambda_2\right|^2 < \ldots < \left|\lambda_{n-2}\right|^2 \leq d \ . \tag{7}$$

Le soluzioni (4) dell'equazione (3) non sono difficili da calcolare numericamente. Dapprima, è necessario determinare i coefficienti $\{\alpha_j\}_{j=0}^n$ del polinomio caratteristico, i quali sono tali che:

$$\mathcal{P}(\beta) = \sum_{j=0}^{n} \alpha_j \beta^j \ . \tag{8}$$

A questo scopo, è conveniente impostare un opportuno sistema lineare (di $n+1$ equazioni in $n+1$ incognite) del tipo

$$A\underline{\alpha} = \underline{b} \ . \tag{9}$$

Ciò può essere fatto convenientemente, valutando il polinomio caratteristico $\mathcal{P}(\beta) = \det(U^T U - \beta\mathbf{I})$ in corrispondenza a $n+1$ valori equidistanziati $\{\tilde{\beta}_j\}_{j=0}^n$ tali che

$$\tilde{\beta}_j = c + j\frac{d-c}{n} \qquad \forall\ j = 0, \ldots n \ ; \tag{10}$$

siano, quindi, la matrice A e il vettore b definiti come segue:

$$A = \begin{pmatrix} \tilde{\beta}_0^0 & \ldots & \tilde{\beta}_0^n \\ \vdots & \vdots & \vdots \\ \tilde{\beta}_n^0 & \ldots & \tilde{\beta}_n^n \end{pmatrix} \quad , \qquad \underline{b} = \begin{pmatrix} \mathcal{P}(\tilde{\beta}_0) \\ \vdots \\ \mathcal{P}(\tilde{\beta}_n) \end{pmatrix} \ . \tag{11}$$

Alcune semplici considerazioni sui massimi e i minimi del polinomio caratteristico $\mathcal{P}(\beta) = \det(U^T U - \beta\mathbf{I})$ permettono di concludere che $|\lambda_0|^2$, $|\lambda_2|^2$, $\ldots$ $|\lambda_{n-2}|^2$ sono soluzioni di molteplicità 1 dell'equazione

$$\mathcal{P}'(\beta) = 0 \ .$$

Inoltre, siccome è facile dimostrare che il polinomio caratteristico $\mathcal{P}(\beta)$ ha sicuramente la concavità rivolta verso l'alto nell'insieme $(-\infty, |\lambda_0|^2]$, allora il *metodo di Newton* converge sicuramente proprio a $|\lambda_0|^2$ se viene applicato utilizzando l'estremo c come approssimazione iniziale. Il *metodo di Newton* può essere adattato al presente contesto così come descritto dal seguente algoritmo:

(i) si fissi una "soglia di tolleranza sull'errore assoluto" $\sigma > 0$;
(ii) si ponga $x = x_0$, dove x_0 rappresenta un'approssimazione iniziale;
(iii) si eseguano ripetutamente le seguenti istruzioni (iv)–(vi) mentre è verificata la condizione espressa al punto (vii);
  - (iv) si calcoli la quantità f_x in modo che essa sia uguale a $\mathcal{P}'(x)$;
  - (v) si calcoli la quantità $\mathrm{d}f$ in modo che essa sia uguale a $\mathcal{P}''(x)$;
  - (vi) si ponga $\mathrm{d}x = -f_x/\mathrm{d}f$ e si riaggiorni la soluzione approssimata x in modo da incrementarla della quantità $\mathrm{d}x$;

(vii) se è soddisfatta la seguente condizione sullo spostamento dell'approssimazione

$$|\mathrm{d}x| > \sigma \ ,$$

allora si torni al punto (iv).
Quando l'algoritmo sarà terminato, allora si intende che x è una soluzione dell'equazione $\mathcal{P}'(x) = 0$, a meno di un errore assoluto non superiore alla soglia di tolleranza σ.
Utilizzando ripetutamente il precedente algoritmo e la divisione polinomiale "alla Ruffini" (dopo che è stata determinata una radice di un'equazione), si possono determinare non solo una, ma tutte le soluzioni (4) dell'equazione (3).

Gli autovalori (e, successivamente, gli autovettori) della matrice di partenza U possono essere determinati, utilizzando in modo opportuno gli autovettori della matrice $U^T U$. L'osservazione fondamentale che ci viene in soccorso è la seguente: siano $\mathcal{V}_{2j}$ e $\mathcal{V}_{2j+1}$ i due autospazi (entrambi di dimensione 1) della matrice U, rispettivamente relativi agli autovalori λ_{2j} e λ_{2j+1}, allora $\mathcal{V}_{2j} \oplus \mathcal{V}_{2j+1}$ è l'autospazio della matrice $U^T U$ relativo a $|\lambda_{2j}|^2$, $\forall\ j = 0, \ldots, n/2 - 1$. Da un punto di vista numerico, due vettori $\underline{e}_{j;0}$ e $\underline{e}_{j;1}$, che costituiscono una base ortonormale dell'autospazio della matrice $U^T U$ relativo a $|\lambda_{2j}|^2$, possono essere determinati adattando il *metodo delle potenze inverse* (così come verrà dettagliatamente descritto nell'obiettivo 5, descritto alla fine di questo testo). Si può facilmente dimostrare che la matrice U esercita due effetti combinati sull'autospazio $\mathcal{V}_{2j} \oplus \mathcal{V}_{2j+1}$: una "dilatazione" di coefficiente $|\lambda_{2j}|$ e una rotazione di angolo ϑ_j. Ciò consente di completare la determinazione di tutti gli autovalori $|\lambda_{2j}| \exp(\pm \mathbf{i} \vartheta_j)$, $\forall\ j = 0, \ldots, n/2 - 1$. Infatti, si tratta semplicemente di determinare l'angolo ϑ_j associato alla matrice di rotazione

$$
\begin{pmatrix} \cos\vartheta_j & \sin\vartheta_j \\ -\sin\vartheta_j & \cos\vartheta_j \end{pmatrix} = \frac{1}{|\lambda_{2j}|} \begin{pmatrix} \underline{e}_{j;0} \cdot U \underline{e}_{j;0} & \underline{e}_{j;0} \cdot U \underline{e}_{j;1} \\ \underline{e}_{j;1} \cdot U \underline{e}_{j;0} & \underline{e}_{j;1} \cdot U \underline{e}_{j;1} \end{pmatrix} . \tag{12}
$$

Obiettivo (intermedio) 1:

si scriva un programma in linguaggio **C** che *legge da un dato file una matrice U e, dopo aver calcolato $U^T U$, determina gli estremi c e d dell'intervallo che include la parte reale dei cerchi di Gershgorin e, quindi, tutti gli autovalori della matrice $U^T U$, così come descritto dalla relazione (6).* Il programma deve contenere:

(A) una *function* che ha tre argomenti: una matrice A, il valore della dimensione effettiva n e, infine, una seconda matrice B; tale *function* restituisce (attraverso il terzo argomento) la trasposta della prima matrice $B = A^T$;

(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) una *function* che ha tre argomenti: una stringa, il valore della dimensione effettiva n e, infine, un vettore; tale *function* restituisce (attraverso il terzo argomento) i valori degli n elementi del vettore che sono stati prelimarmente scritti in modo ordinato nella stringa;

(D) una *function* che ha quattro argomenti: i primi due sono una matrice A e il valore della sua dimensione effettiva n; tale *function* restituisce (attraverso il terzo e il quarto argomento) i valori degli estremi c e d definiti dalle formule (5) e (6), dove qui si intende che $\{a_{i,j}\}_{i,j=0}^{n-1}$ sono i coefficienti della matrice A corrispondente al primo argomento della *function*;

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

(E1) *l'apertura di un file di input* che si chiama `autoval_compl_con_a_coppie.inp`, il quale deve essere preliminarmente scaricato dalla rete e posizionato nella stessa cartella contenente il programma che stiamo descrivendo; i dati relativi a una matrice U, tale che i suoi autovalori soddisfano le proprietà descritte in (1), sono appunto contenuti nel file `autoval_compl_con_a_coppie.inp`;

(E2) la lettura dal *file di input* della dimensione effettiva n delle matrici; si effettuino dei *test*, in modo tale che se $n < 4$, o se n è dispari oppure se $n > NDIM$ (si veda poi per la definizione della dimensione massima $NDIM$), allora l'esecuzione del programma deve essere immediatamente arrestata;

(E3) la lettura dal *file di input* delle n righe che compongono la matrice U; la definizione dei valori numerici di tutti gli elementi di U deve essere effettuata grazie a n opportune *chiamate della function descritta al punto (C)*;

(E4) il calcolo della matrice $U^T U$ tramite due opportune *chiamate delle function descritte ai punti (A) e (B)*;

(E5) la stampa sul video dei coefficienti della matrice $U^T U$ in modo che sia *esteticamente non disprezzabile*;

(E6) il calcolo degli estremi c e d tramite un'opportuna *chiamata della function descritta al punto (D)*;

(E7) la stampa sul video degli estremi c e d in modo che sia *esteticamente non disprezzabile.*

Alcuni consigli

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).

È sicuramente opportuno stabilire il valore *costante* di $NDIM$ per mezzo di una direttiva `#define`. Tale valore di $NDIM$ sia fissato in modo che sia *dispari.*

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 delle componenti delle matrici verranno effettuate tenendo conto che la *dimensione effettiva* dello spazio vettoriale che stiamo considerando è $n \leq NDIM$.

Obiettivo (intermedio) 2:

si integri il programma richiesto dall'obiettivo 1, in modo tale da calcolare tutti i coefficienti $\{\alpha_j\}_{j=0}^{n}$ che compaiono nell'espansione (8) del polinomio caratteristico $\mathcal{P}(\beta) = \det(U^T U - \beta \mathbf{I})$. A tal fine si proceda come segue:

(A) si includano tutte le *function* necessarie per il calcolo del determinante di una matrice;

(B) si scriva una *function* che restituisce il valore del polinomio caratteristico $\mathcal{P}(\lambda) = \det(S - \lambda \mathbf{I})$ e ha tre argomenti: una matrice S, il valore della variabile λ e il valore della dimensione effettiva n; tale *function*, *al suo interno* definisce una matrice temporanea, i cui coefficienti devono essere tali che essa sia uguale a $S - \lambda \mathbf{I}$; inoltre, tale *function* esegue il calcolo di $\det(S - \lambda\mathbf{I})$, tramite un'opportuna chiamata ad una delle function richieste al punto (A), e infine restituisce il valore così calcolato all'ambiente chiamante;

(C) si includa una *function* che effettua il calcolo della soluzione di un sistema lineare di n equazioni in n incognite del tipo $A\underline{x} = \underline{b}$, dove (quando appaiono in argomento) $\underline{b}$ e $\underline{x}$ sono vettori $NDIM$–dimensionali e A una matrice $NDIM \times NDIM$–dimensionale;

(D) all'interno della *main function*, si scrivano gli opportuni cicli che consentono di definire tutti gli elementi della matrice $(n+1) \times (n+1)$–dimensionale A e del vettore noto $\underline{b}$, in accordo con le definizioni riportate nelle formule (10)–(11); ovviamente, il calcolo degli elementi del vettore noto $\underline{b}$ deve essere effettuato tramite delle opportune *chiamate della function descritta al punto (B)*;

(E) all'interno della *main function*, si determini il vettore di coefficienti $\underline{\alpha}$ che risolve l'equazione $A\underline{\alpha} = \underline{b}$ tramite un'opportuna *chiamata della function descritta al punto (C)*; (attenzione! *Si ricorda che questo sistema è di* $n+1$ *equazioni in* $n+1$ *incognite*);

(F) all'interno della *main function*, si effettui la stampa sul video dei coefficienti del vettore $\underline{\alpha}$, in modo che sia esteticamente non disprezzabile.

Obiettivo (intermedio) 3:

si integri il programma richiesto dall'obiettivo 2, in modo tale da determinare il più piccolo degli autovalori (cioè $|\lambda_0|^2$) della matrice $U^T U$. A tal fine si proceda come segue:

(A) si scriva una *function* che ha tre argomenti: un vettore di coefficienti $\underline{\alpha}$ di un polinomio $\mathcal{P}(x)$, il valore del grado n di tale polinomio, il valore della variabile x; tale *function* restituisce il valore della derivata $\mathcal{P}'(x)$, dove $\mathcal{P}(x) = \sum_{j=0}^{n} \alpha_j x^j$;

(B) si scriva una *function* che ha tre argomenti: un vettore di coefficienti $\underline{\alpha}$ di un polinomio $\mathcal{P}(x)$, il valore del grado n di tale polinomio, il valore della variabile x; tale *function* restituisce il valore della derivata seconda $\mathcal{P}''(x)$, dove $\mathcal{P}(x) = \sum_{j=0}^{n} \alpha_j x^j$;

(C) si scriva una *function* che ha quattro argomenti: un vettore di coefficienti $\underline{\alpha}$ di un polinomio $\mathcal{P}(x)$, il valore del grado n di tale polinomio, il valore di un'approssimazione iniziale x_0 e una "soglia di tolleranza sull'errore assoluto" σ ; tale *function* restituisce il valore (approssimato) di una soluzione x dell'equazione $\mathcal{P}'(x) = 0$, dove $\mathcal{P}(x) = \sum_{j=0}^{n} \alpha_j x^j$; tale soluzione deve essere determinata applicando il *metodo di Newton*, seguendo l'algoritmo descritto ai punti (i)–(vii) precedenti; ovviamente, all'interno di tale *function* (che viene descritta in questo punto (C)) dovranno comparire delle opportune *chiamate delle function descritte ai punti (A) e (B)*;

(D) all'interno della *main function*, si effettui il calcolo dell'autovalore $|\lambda_0|^2$, tramite un'opportuna *chiamata della function descritta al punto (C)*, con valori passati in argomento in modo che l'approssimazione iniziale sia fissata uguale a c e la "soglia di tolleranza sull'errore assoluto" sia posta uguale a $2 \cdot 10^{-16}\, d$;

(E) all'interno della *main function*, si effettui la stampa sul video del valore calcolato di $|\lambda_0|^2$, così come richiesto al punto (D), in modo che sia esteticamente non disprezzabile.

Obiettivo (intermedio) 4:

si integri il programma richiesto dall'obiettivo 2, in modo tale da determinare tutti gli autovalori (cioè $|\lambda_0|^2$, $|\lambda_2|^2$, $\ldots$ $|\lambda_{n-2}|^2$) della matrice $U^T U$. A tal fine si proceda come segue:

(A) si scriva una *function* che ha tre argomenti: un vettore di coefficienti $\underline{p}$ di un polinomio $P(x) = \sum_{j=0}^{n} p_j x^j$, il valore del grado n di tale polinomio, il valore di una radice r del polinomio, cioè $P(r) = \sum_{j=0}^{n} p_j r^j = 0$; *alla fine della chiamata, tale function* dovrà restituire, proprio attraverso il primo dei suoi argomenti i coefficienti q_0 , ... q_{n-1} del polinomio quoziente $Q(x) = \sum_{j=0}^{n-1} q_j x^j$ tale che $P(x) = (x - r)Q(x)$; il calcolo di tali coefficienti può essere svolto seguendo il breve algoritmo descritto ai seguenti punti (A1)–(A2), che implementano parte del ben noto *metodo di Ruffini*;

(A1) si scriva un ciclo che deve essere ripetuto, mentre un indice i procede a ritroso da $n - 1$ a 0 ; all'interno di questo ciclo, il valore di ogni coefficiente p_i dovrà essere incrementato della quantità $r p_{i+1}$;

(A2) si scriva un ciclo che deve essere ripetuto, mentre un indice i procede da 0 a $n-1$; all'interno di questo ciclo, "si abbassa di uno il grado", cioè si pone $p_i = p_{i+1}$;

(B) all'interno della *main function*, si effettui il calcolo e la stampa (ordinata) su video di tutti gli autovalori $|\lambda_0|^2$, $|\lambda_2|^2$, ... $|\lambda_{n-2}|^2$; a tale scopo, si scriva un ciclo che deve essere ripetuto, mentre un indice j procede da 0 a $n/2-1$; all'interno di tale ciclo l'autovalore $|\lambda_{2j}|^2$ deve essere determinato tramite *un'opportuna chiamata della function descritta al punto (C) dell'obiettivo 3, in modo simile a quanto richiesto al punto (D) di quello stesso obiettivo 3*; (ma attenzione! *Si osservi che il metodo si dovrà applicare a "un polinomio caratteristico ridotto" di grado* $n - 2j$); inoltre, all'interno di tale ciclo si deve "ridurre la doppia radice $|\lambda_{2j}|^2$ del polinomio caratteristico" $\mathcal{P}(\lambda) = \det(S - \lambda \mathbf{I})$, grazie a *due opportune chiamate della function descritta al punto (A).*

Obiettivo (finale) 5:

si integri il programma richiesto dall'obiettivo 4, in modo tale da calcolare la matrice di rotazione associata a ogni coppia coniugata di autovalori λ_{2j} , λ_{2j+1} , $\forall\ j = 0, \ldots, n/2-1$. A tal fine si proceda come segue:

(A) si includa nel programma 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 includa nel programma una *function* che ha tre argomenti: una matrice A , il valore della dimensione effettiva n e una seconda matrice *inv_A*; tale *function* restituisce (attraverso il terzo argomento) la matrice inversa $inv\_A = A^{-1}$;

(E) si scriva una *function* che ha cinque argomenti: una matrice A (che si intende essere simmetrica e con $n/2$ autovalori ciascuno di molteplicità 2), il valore della dimensione effettiva n , un'approssimazione γ di uno degli autovalori e una coppia di vettori $\underline{e}_0$, $\underline{e}_1$; tale *function* restituisce (attraverso il quarto argomento e il quinto argomento) due autovettori ortonormali relativi all'autovalore ben approssimato da γ ; il calcolo deve essere effettuato applicando il *metodo delle potenze inverse*, descritto come ai seguenti punti (E1)–(E4); ovviamente, per poter tradurre in linguaggio **C** le prescrizioni contenute nei seguenti punti (E1)–(E4), laddove è conveniente, dovranno essere effettuate le opportune *chiamate delle function descritte ai precedenti punti (A)–(D)*;

(E1) si introducano le seguenti definizioni come approssimazioni iniziali dell'autovalore e degli autovettori: $\lambda = -10^{15}$, $\underline{e}_0 = \big(1\,,\,0,\,\ldots\,,\,0\big)$, $\underline{e}_1 = \big(0,\,1\,,\,0,\,\ldots\,,\,0\big)$;

(E2) si calcoli $(A - \gamma\mathbf{I})^{-1}$;

(E3) si iterino le operazioni descritte ai seguenti punti (E31)–(E35) mentre è verificata la condizione (di permanenza nel ciclo), che è espressa al punto (E4);

(E31) si ponga $\overline{\lambda} = \lambda$;

(E32) si (ri)calcolino i valori degli elementi dei vettori $\underline{w}_0 = (A - \gamma\mathbf{I})^{-1}\underline{e}_0$ e $\underline{w}_1 = (A - \gamma\mathbf{I})^{-1}\underline{e}_1$;

(E33) si (ri)ortogonalizzino $\underline{w}_0$ e $\underline{w}_1$, ridefinendo $\underline{w}_1 = \underline{w}_1 - (\underline{w}_0 \cdot \underline{w}_1)\underline{w}_0/(\|\underline{w}_0\|^2)$;

(E34) si ricalcoli l'approssimazione dell'autovalore, in modo da porre $\lambda = (\underline{e}_0 \cdot \underline{w}_0 + \underline{e}_1 \cdot \underline{w}_1)/2$;

(E35) si ridefiniscano i vettori $\underline{e}_0$ e $\underline{e}_1$ in modo tale che

$$\underline{e}_0 = \frac{\underline{w}_0}{\|\underline{w}_0\|}\ , \qquad \underline{e}_1 = \frac{\underline{w}_1}{\|\underline{w}_1\|}\ ;$$

(E4) se l'errore relativo riguardante la determinazione dell'autovalore è significativamente maggiore dell'*errore di macchina*, cioè se si verifica che

$$\frac{|\lambda - \bar{\lambda}|}{|\lambda|} > n\ \times 10^{-15}\ ,$$

allora si torni a ripetere le operazioni descritte ai precedenti punti (E31)–(E35);

(F) all'interno della *main function*, si ponga $\Delta = \min_{j=1,\ldots,n/2-1}\{|\lambda_{2j}|^2 - |\lambda_{2j-2}|^2\}$ e si scriva un ciclo che deve essere ripetuto, mentre un indice j procede da 0 a $n/2 - 1$; all'interno di tale ciclo, gli autovettori $\underline{e}_{j;0}$ e $\underline{e}_{j;1}$ devono essere determinati con un'opportuna *chiamata della function descritta al punto (E)*, utilizzando come terzo argomento il valore di $|\lambda_{2j}|^2 - \Delta/20$; infine, sempre all'interno di tale ciclo, si stampi sul video la matrice di rotazione definita dal secondo membro dell'equazione (12).