Annamaria Mazzia

Appendice di esercizi in FORTRAN

Dipartimento di Ingegneria Civile Edile e Ambientale
Università degli Studi di Padova

a.a. 2012/2013

Appendice A

Esercizi di programmazione FORTRAN

Es. 1 — *(appello del 13 febbraio 2013)* Si vuole calcolare la traccia delle matrici P, Q e R, dove P è una matrice di dimensione $m \times m$, Q è una matrice costruita a partire dalla P applicando la formula $Q = P + \beta I$ con β parametro reale e I matrice identità $(m \times m)$ mentre $R = PQ$. Si scriva, dunque, un programma FORTRAN che

1. legge la dimensione m, la matrice P e il parametro β;
   costruisce la matrice $Q = P + \beta I$ senza fare uso esplicito della matrice identità;
3. costruisce la matrice $R = PQ$ utilizzando la subroutine MATRMATR;
4. calcola e stampa la traccia della matrice P
5. calcola e stampa la traccia della matrice Q
6. calcola e stampa la traccia della matrice R.
   Per calcolare la traccia di una matrice, si usi la function TRACCIA (si ricorda che $tr(P) = \sum_{i=1}^{m} p_{ii}$).

Es. 2 — *(appello del 18 settembre 2012)* Si scriva un programma FORTRAN che implementa la fattorizzazione di Cholesky secondo il seguente algoritmo:

$$
\begin{aligned}
&\forall j = 1, 2, \ldots, n \\
&\qquad l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2} \\
&\qquad \forall i = j+1, j+2, \ldots, n \\
&\qquad\qquad l_{ij} = \frac{1}{l_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right)
\end{aligned}
$$

Il programma dovrà leggere da file la dimensione $(n \leq 25)$ la matrice A, chiamare la subroutine CHOL che implementa l'algoritmo sopra descritto e quindi stampare a file la matrice triangolare inferiore L.

Es. 3 — *(appello del 4 settembre 2012)* Si vuole risolvere l'equazione: $f(x) = x^2 - \cos(x) = 0$ con lo schema iterativo

$$x_{n+1} = x_n - f(x_n)/k_0.$$

Scrivere un programma fortran che:

1. Legge la soluzione iniziale x_0, il valore k_0, il numero massimo di iterazioni $ITMAX$ e la tolleranza *tol*;

2. implementa lo schema iterativo proposto arrestando l'esecuzione o quando $|x_{n+1} - x_n| < tol$ oppure quando si è raggiunto il numero massimo di iterazioni ITMAX; ad ogni iterazione $n \geq 1$ memorizza l'approssimazione x_n nella componente omonima del vettore SOL;
3. stampa la soluzione finale e il numero di iterazioni eseguite;
4. calcola, impiegando la function EUC, la norma euclidea α del vettore SOL; stampa il valore α.

Es. 4 — *(appello del 10 luglio 2012)* Si vuole effettuare un passo del metodo dello *Steepest Descent* per approssimare la soluzione del sistema lineare $B\mathbf{x} = \mathbf{c}$ con B matrice quadrata, simmetrica, definita positiva di dimensione n. Partendo da un'approssimazione iniziale $\mathbf{x}_0$, questo metodo approssima la soluzione iterando l'algoritmo

$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\mathbf{r}_k$ dove $\mathbf{r}_k$ è il residuo al passo k mentre $\alpha = \dfrac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{r}_k^T B\mathbf{r}_k}$.

Si scriva dunque un programma FORTRAN che:

1.

   legge la dimensione n, la matrice B e i vettori $\mathbf{c}$ e $\mathbf{x}_0$;
3. calcola il residuo $\mathbf{r}_0 = \mathbf{c} - B\mathbf{x}_0$;
4. calcola $\delta_1 = \mathbf{r}_0^T\mathbf{r}_0$;
5. calcola $\mathbf{q} = B\mathbf{r}_0$
6. calcola $\delta_2 = \mathbf{r}_0^T\mathbf{q}$;
7. calcola $\alpha = \dfrac{\delta_1}{\delta_2}$;
8. calcola e stampa $\mathbf{x} = \mathbf{x}_0 + \alpha\mathbf{r}_0$; Dove occorre, si usino la subroutine PRODAX che effettua il prodotto matrice vettore e la function DOTS che effettua il prodotto scalare tra due vettori.

Es. 5 — *(appello del 26 giugno 2012)* Si scriva un programma Fortran che, a partire da una matrice quadrata A di dimensione al più 25 e i vettori $\mathbf{b}$ e $\mathbf{x}$ calcoli un passo del metodo di Jacobi $y = E\mathbf{x} + \mathbf{q}$: Il programma Fortran dovrà dunque

1. Leggere la dimensione n, la matrice A i vettori $\mathbf{b}, \mathbf{x}$.
2. Calcolare E mediante la subroutine `JacobiIter`;
3. Calcolare $\mathbf{q}$;
4. Calcolare $\mathbf{v} = E\mathbf{x}$ invocando la subroutine `APERX`;
5. Calcolare il vettore risultato $\mathbf{y} = \mathbf{v} + \mathbf{q}$; stampare a video il vettore $\mathbf{y}$. Si devono inoltre mettere a punto i sottoprogrammi `JacobiIter` e `APERX`. **Nota**. La matrice di iterazione $E = -D^{-1}(L + U)$ deve essere calcolata per componenti:

$$e_{ij} = \begin{cases} 0 & \text{se } i = j \\ -\dfrac{a_{ij}}{a_{ii}} & \text{se } i \neq j \end{cases}, \quad i = 1, \dots, n, \quad j = 1, \dots, n$$

Il vettore $\mathbf{q} = D^{-1}\mathbf{b}$ si deve calcolare per componenti, ovvero $q_i = \dfrac{b_i}{a_{ii}}, \qquad i = 1, \dots, n.$

Es. 6 — *(appello del 15 febbraio 2012)* Data la matrice quadrata A di dimensione $n \leq 40$, la matrice diagonale D contenente i termini diagonali di A e due vettori $\mathbf{y}$ e $\mathbf{b}$ della medesima dimensione, si vuola calcolare un passo dello schema di Jacobi come:

$$\mathbf{x} = (I - D^{-1}A)\mathbf{y} + D^{-1}\mathbf{b}$$

Scrivere un programma FORTRAN che calcoli $\mathbf{x}$ eseguendo le seguenti operazioni:

1. legge $n, A, \mathbf{b}$ e $\mathbf{y}$;

2. calcola il vettore $\mathbf{q} = D^{-1}\mathbf{b}$ mediante la subroutine INVD;
3. calcola il vettore $\mathbf{y} = A\mathbf{y}$ mediante la subroutine MATVET;
4. calcola il vettore $\mathbf{v} = D^{-1}\mathbf{u}$ mediante la subroutine INVD;
5. calcola il vettore $\mathbf{x} = \mathbf{y} - \mathbf{v} + \mathbf{q}$;
6. stampa $\mathbf{x}$ con commento.

Es. 7 — *(appello del 15 settembre 2011)* Data la matrice quadrata A di dimensione $n \leq 50$ e i fattori triangolari F e F^T, basso e alto rispettivamente, tali che $FF^T = (\tilde{A})^{-1}$ con $\tilde{A}$ un'approssimazione di A, si vuole misurare la lunghezza euclidea del vettore differenza $(\mathbf{v} - \mathbf{x})$ ove $\mathbf{x}$ è un vettore assegnato e:

$$\mathbf{v} = FAF^T\mathbf{x}$$

Scrivere un programma in linguaggio FORTRAN che:

1. legge n, A, F e $\mathbf{x}$;
2. calcola $G = F^T$
3. calcola il vettore $\mathbf{y} = G\mathbf{x}$ con la subroutine MATVET;
4. calcola il prodotto $\mathbf{z} = A\mathbf{y}$ con la subroutine MATVET;
5. calcola il vettore $\mathbf{v} = F\mathbf{z}$ con la subroutine MATVET;
6. calcola il vettore differenza $\mathbf{w} = \mathbf{v} - \mathbf{x}$;
7. calcola la norma euclidea di $\mathbf{w}$ mediante la function NORMA2 e la stampa.

Es. 8 — *(appello del 1º settembre 2011)* Data la matrice simmetrica e definita positiva A di dimensione $N \leq 50$, calcolare l'autovalore massimo con il metodo del quoziente di Rayleigh:

$$\lambda = \lim_{k\to\infty} r_k \text{ ove } r_k = \frac{\mathbf{x}_k^T A\mathbf{x}_k}{\mathbf{x}_k^T\mathbf{x}_k},\ \mathbf{x}_{k+1} = A\mathbf{x}_k$$

Scrivere un programma FORTRAN che esegua nell'ordine le seguenti operazioni:

1. legge la dimensione N, la matrice A, il numero massimo di iterazioni $ITMAX$, la tolleranza TOL;
2. costruisce il vettore iniziale $\mathbf{x}_0$ con tutte le componenti uguali a 1;
3. ad ogni iterazione:
   - calcola il vettore $\mathbf{y}_k = A\mathbf{x}_k$ mediante la subroutine MATV;
   - calcola gli scalari $\alpha_k = \mathbf{x}_k^T\mathbf{y}_k$ e $\beta = \mathbf{x}_k^T\mathbf{x}_k$ impiegando in entrambi i casi la function SCAL;
   - calcola il rapporto $r_k = \alpha_k/\beta_k$;
   - calcola lo scarto $d_k = |r_k - r_{k-1}|$;
   - aggiorna $\mathbf{x}_k$ e r_{k-1} per l'iterazione successiva;
4. arresta le iterazioni quando si è raggiunto $ITMAX$ oppure $d_k < TOL$;
5. stampa l'autovalore massimo $\lambda = r_k$ e il numero di iterazioni eseguite.

Es. 9 — *(appello del 5 luglio 2011)* Assegnate due matrici quadrate P e Q di dimensione m ($m \leq 80$), si calcoli la norma di Frobenius (o euclidea) della matrice $A = I + PQ + QP$ (dove I è la matrice identità), vale a dire $\alpha = \sqrt{\sum_{i,j=1}^n A_{ij}^2}$.

Scrivere dunque un programma in linguaggio FORTRAN che:

1. legge m, P e Q;
2. calcola le matrici $V = PQ$ e $Z = QP$ servendosi in maniera opportuna della subroutine MATRMATR che esegue il prodotto di due matrici quadrate;
3. calcola la matrice $A = I + V + Z$;
4. calcola α (la norma di Frobenius della matrice A) servendosi della function FROBENIUS;

5. stampa il risultato ottenuto.

Es. 10 — *(appello del 21 giugno 2011)* Si vuole risolvere il sistema lineare $A\mathbf{x} = \mathbf{b}$ di dimensione $m \leq 40$ mediante il metodo iterativo di *Richardson*:

$$\mathbf{x}_{k+1} = (I - \delta A)\mathbf{x}_k + \delta \mathbf{b}$$

con δ numero reale positivo. Scrivere un programma in linguaggio FORTRAN che

1. legge $m, A, \mathbf{b}, \delta$ la soluzione iniziale $\mathbf{x}_0$, la tolleranza sullo scarto $TOLL$ e il numero massimo di iterazioni $ITMAX$;
2. ad ogni iterazione calcola il vettore $\mathbf{v} = \delta A \mathbf{x}_k$ mediante la subroutine $DMATVET$, la soluzione corrente $\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{v} + \delta \mathbf{b}$ e lo scarto $\mathbf{d}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_k$;
3. calcola la norma euclidea del vettore scarto $\|\mathbf{d}_{k+1}\|_2$ mediante la function $NORM2$;
4. arresta le iterazioni quando $\|\mathbf{d}_{k+1}\|_2 \leq TOLL$ oppure il numero di iterazioni supera $ITMAX$;
5. stampa la soluzione ed il numero di iterazioni effettuate.

Es. 11 — *(primo compitino del 28 aprile 2011)* Si vuole risolvere l'equazione non lineare $f(x) = 0$ con $f(x) = \sqrt{x^3 - 2} - \ln(x^2)$, mediante il seguente schema iterativo noto come *schema di Steffensen*:

$$x_{k+1} = x_k - \frac{h_k^2}{f(x_k + h_k) - h_k}$$

in cui $h_k = f(x_k)$. Scrivere un programma in linguaggio FORTRAN che:

1. legge la soluzione iniziale x_0, la tolleranza sullo scarto τ e il numero massimo di iterazioni $ITMAX$;
2. implementa lo schema assegnato arrestando le iterazioni quando $|x_{k+1} - x_k| \leq \tau$ oppure il numero massimo di iterazioni supera $ITMAX$;
3. stampa la soluzione ed il numero di iterazioni effettuate.
   Si utilizzi una function per il calcolo di $f(x)$.

Es. 12 — *(appello del 16 febbraio 2011)* Date due matrici quadrate reali A e B di dimensione n ($n \leq 50$), si calcoli la norma euclidea (o norma di Frobenius) α della matrice $C = AB$. Scrivere un programma FORTRAN che:

1. legge e stampa n, A, e B;
2. calcola la matrice prodotto $C = AB$ utilizzando la subroutine PRODMAT;
3. calcola $\alpha = \sqrt{\sum_{i,j=1}^{n} c_{ij}^2}$ utilizzando la function FROB;
4. stampa il valore di α con commento.

Es. 13 — *(appello del 16 settembre 2010)* Si vuole risolvere il seguente schema iterativo

$$\mathbf{y}_0 = \mathbf{z}$$
$$\mathbf{y}_{k+1} = (I - B)\mathbf{y}_k + \mathbf{z}$$

dove I è la matrice identità e B una matrice quadrata, di dimensione m ($m \leq 40$), e $\mathbf{z}$ è un vettore di lunghezza m.

Scrivere un programma in linguaggio FORTRAN che:

1. legge m, B, $\mathbf{z}$, la tolleranza tol e il numero massimo di iterazioni $MAXIT$;
2. pone $\mathbf{y}_0 = \mathbf{z}$;
3. calcola $C = I - B$;
4. implementa lo schema assegnato andando avanti con le iterazioni fino a quando la norma assoluta (o norma 1) della quantità $\mathbf{r}_{k+1} = \mathbf{y}_{k+1} - \mathbf{y}_k$ diventa minore di tol o il numero di iterazioni diventa superiore a $MAXIT$;

5. stampa la soluzione approssimata $\mathbf{y}_{k+1}$ e il numero di iterazioni effettuate.
   Avvalersi della subroutine MATV per il prodotto matrice-vettore e della function NORMA1 per il calcolo della norma assoluta.

Es. 14 — *(appello del 2 settembre 2010)* Date due matrici quadrate A e B di dimensione n ($n \le 40$) e il vettore $\mathbf{x}$ di dimensione n si vuole calcolare il quoziente $\gamma = \dfrac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T B\mathbf{x}}$; scrivere un programma FORTRAN che:

1. legge n, le matrici A e B e il vettore $\mathbf{x}$;
2. calcola i due vettore $\mathbf{y} = A\mathbf{x}$ e $\mathbf{z} = B\mathbf{x}$ servendosi della subroutine MATVET che esegue il prodotto matrice vettore;
3. calcola $\alpha = \mathbf{x}^T\mathbf{y}$ e $\beta = \mathbf{x}^T\mathbf{z}$ impiegando la function SCAL che esegue il prodotto scalare di due vettori;
4. calcola e stampa $\gamma = \alpha/\beta$

Es. 15 — *(appello del 7 luglio 2010)* Date le tre matrici quadrate S, T, V di dimensione n ($n \le 80$) si vuole calcolare la matrice $B = STV$ e trovarne la traccia.

Scrivere un programma in linguaggio FORTRAN che:

1. legge n, S, T , V
2. calcola prima la matrice $A = TV$ e successivamente la matrice $B = SA$ (impiegare in entrambi i casi la subroutine MATRMATR che esegue il prodotto tra due matrici)
3. calcola la traccia β della matrice B (impiegare la function TRACCIA)
4. stampa β con commento.

Es. 16 — *(appello del 16 giugno 2010)* Data una matrice quadrata A e i vettori $\mathbf{x}$ e $\mathbf{r}$ di dimensione $n \le 50$, si vuole calcolare il vettore $\mathbf{y}$:

$$\mathbf{y} = \mathbf{x} + \alpha\mathbf{r}$$

dove $\alpha = \mathbf{r}^T\mathbf{r}/\mathbf{r}^T A\mathbf{r}$. Scrivere un programma FORTRAN che:

1. legge $n, A, \mathbf{x} e \mathbf{r}$,
2. calcola il vettore $\mathbf{z} = A\mathbf{r}$,
3. calcola il prodotto scalare $d = \mathbf{r}^T\mathbf{z}$,
4. calcola il prodotto scalare $c = \mathbf{r}^T\mathbf{r}$,
5. calcola lo scalare $\alpha = c/d$,
6. calcola e stampa il vettore $\mathbf{y} = \mathbf{x}+\alpha\mathbf{r}$, utilizzando la subroutine MATVET per il prodotto matrice-vettore e la function SCAL per il prodotto scalare.

Es. 17 — *(compitino del 15 aprile 2010)* Si vuole risolvere l'equazione non lineare

$$f(x) = \sqrt{x^2+3} - e^{5x} = 0$$

con il seguente schema iterativo:

$$x_{k+1} = x_k - Cf(x_k)$$

dove C è un coefficiente scalare costante. Scrivere un programma FORTRAN che:

1. legge la soluzione iniziale x_0, la costante C, la tolleranza sullo scarto τ e il numero massimo di iterazioni $ITMAX$;
2. implementa lo schema assegnato arrestando le iterazioni quando $|x_{k+1} - x_k| \le \tau$ oppure il numero di iterazioni supera $ITMAX$;
3. stampa la soluzione ed il numero di iterazioni effettuate.
   Si utilizzi una function per il calcolo di $f(x)$.

Es. 18 — *(appello del 18 settembre 2009)* Dato il sistema di n equazioni $A\mathbf{u} = \mathbf{b}$ ed una sua soluzione approssimata $\mathbf{w}$, si vuole calcolare il corrispondente residuo relativo $r_r = \|\mathbf{b} - A\mathbf{w}\|_2 / \|\mathbf{b}\|_2$. Scrivere un programma in linguaggio FORTRAN che, utilizzando una subroutine per il prodotto matrice-vettore ed una function per la norma euclidea:

1. legge e stampa $n, A, \mathbf{b}$ e $\mathbf{w}$;
2. calcola il vettore residuo $\mathbf{r} = \mathbf{b} - A\mathbf{w}$;
3. calcola la norma euclidea di $\mathbf{r}$ e $\mathbf{b}$;
4. calcola e stampa r_r.

Es. 19 — *(appello del 2 luglio 2009)* Scrivere un programma in linguaggio FORTRAN che implementi il primo passo dell'algoritmo di Lanczos per la generazione di una base di vettori ortonormali in $\mathbb{R}^n$. Data la matrice simmetrica A ed i vettori $\mathbf{u}_0$ e $\mathbf{u}_1$ di dimensione n ($n \le 25$), si calcoli $\mathbf{u}_2$ come:

$$\mathbf{v} = A\mathbf{u}_1 - \alpha\mathbf{u}_0, \quad \mathbf{u}_2 = \mathbf{v} / \|\mathbf{v}\|_2$$

con $\alpha = \mathbf{u}_1^T A\mathbf{u}_1$ e $\beta = \mathbf{u}_0^T A\mathbf{u}_1$. Il programma deve eseguire le seguenti operazioni:

1. legge da file $n, A, \mathbf{u}_0 e \mathbf{u}_1$;
2. calcola il vettore $\mathbf{w} = A\mathbf{u}_1$;
3. calcola il prodotto scalare $\alpha = \mathbf{u}_1^T\mathbf{w}$;
4. calcola il prodotto scalare $\beta = \mathbf{u}_0^T\mathbf{w}$;
5. calcola il vettore $\mathbf{v} = \mathbf{w} - \alpha\mathbf{u}_1 - \beta\mathbf{u}_0$;
6. calcola la norma euclidea $\gamma = \|\mathbf{v}\|_2$;
7. calcola e stampa $\mathbf{u}_2 = \mathbf{v}/\gamma$.

   Si utilizzi la subroutine PRODMV per il calcolo del prodotto matrice-vettore e le functions DOT e NORM per il calcolo del prodotto scalare e della norma euclidea, rispettivamente.

Es. 20 — *(appello del 6 febbraio 2009)* Date le matrici quadrate A e B di dimensione n ($n \le 30$) si vuole calcolare la traccia della matrice $E = AB - BA$. Scrivere un programma in linguaggio FORTRAN che

1. legge e stampa n, A, B;
2. costruisce la matrice E richiamando due volte la subroutine PRODMAT che esegue il prodotto fra due matrici quadrate;
3. calcola la traccia della matrice E servendosi della function TRAC; stampa con commento il risultato ottenuto.

Es. 21 — *(appello del 18 giugno 2008)* Data la matrice quadrata A di dimensione n ($n \le 25$) e il vettore $\mathbf{x}$ di dimensione n, si vuole calcolare la norma assoluta del vettore $\mathbf{y} = (I + A)\mathbf{x}$. Scrivere un programma in linguaggio FORTRAN che

1. legge e stampa $n, A, \mathbf{x}$;
2. calcola la matrice $B = I + A$ mediante la subroutine SOMIDEN;
3. calcola il vettore $\mathbf{y} = B\mathbf{x}$ mediante la subroutine MVET;
4. calcola la norma assoluta $\|\mathbf{y}\|_1 = \sum_{i=1}^n |y_i|$ mediante la function NASS;
5. stampa con commento la norma trovata.

Es. 22 — *(appello del 13 dicembre 2007)* Data la matrice quadrata reale A ed il vettore reale $\mathbf{x}$ di dimensione n ($n \le 50$), si calcoli la norma assoluta del vettore $\mathbf{y} = A^T\mathbf{x}$. Scrivere un programma FORTRAN che:

1. legge e stampa con commento $n, A, \mathbf{x}$;
2. calcola la matrice $B = A^T$;
3. calcola il vettore $\mathbf{y} = B\mathbf{x}$;
4. calcola $\|\mathbf{y}\|_1 = \sum_{i=1}^n |y_i|$;

5. stampa il valore di $\|\mathbf{y}\|_1$ con commento.

   Si utilizzino le subroutine TRASP e MATVET per calcolare rispettivamente la trasposta di una matrice quadrata e il prodotto matrice-vettore, e la function NASS per calcolare la norma assoluta di un vettore.

Es. 23 — *(appello del 30 agosto 2007)* Una matrice quadrata A di dimensione n ($n \leq 30$) si dice diagonalmente dominante in senso stretto se verifica la condizione:

$$\sum_{k=1,k\neq i}^{n} |a_{ik}| < |a_{ii}| \quad \forall i = 1,2,\ldots,n$$

Scrivere un programma FORTRAN che:

1. legge n e la matrice A; stampa i dati letti con commento;
2. costruisce il vettore $\mathbf{t}$ di dimensione n tale che

$$t_i = \sum_{k=1,k\neq i}^{n} |a_{ik}|$$

3. se per ogni $i = 1,2,\ldots,n$ è verificata la condizione $t_i < |a_{ii}|$ stampa il messaggio seguente: `Matrice diagonalmente dominante`; in caso contrario stampa $i, t_i, |a_{ii}|$ ed interrrompe l'esecuzione.

Es. 24 — *(appello del 20 giugno 2007)* Date le matrici quadrate A e B di dimensione n ($n \leq 50$) e i vettori $\mathbf{x}$ e $\mathbf{y}$ di dimensione n, si vuole calcolare il vettore $\mathbf{z} = A\mathbf{x} + B\mathbf{y}$. Scrivere un programma FORTRAN che

1. acquisisce valori per $n, A, B, \mathbf{x}, \mathbf{y}$; stampa i dati letti con commento;
2. costruisce i vettori $A\mathbf{x}$ e $B\mathbf{y}$ richiamando in entrambi i casi la subroutine PROD che esegue il prodotto fra una matrice e un vettore;
3. calcola e stampa il vettore $\mathbf{z}$.

Es. 25 — *(appello del 31 agosto 2006)* Dato il sistema lineare $A\mathbf{x} = \mathbf{b}$ e una soluzione approssimata $\mathbf{x}_k$, si vuole calcolare il vettore residuo $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ e la sua norma euclidea. Scrivere un programma FORTRAN che

1. legge la dimensione n della matrice A ($n \leq 40$), la matrice A, il termine noto $\mathbf{b}$ e l'approssimazione $\mathbf{x}_k$, stampa i dati letti con commento;
2. calcola il vettore residuo $\mathbf{r}_k$ impiegando la subroutine RESIDUO;
3. calcola la norma euclidea di $\mathbf{r}_k$ utilizzando la function EUCL;
4. stampa la norma trovata.

Es. 26 — *(appello del 23 giugno 2006)* Siano assegnati due vettori $\mathbf{x}$ e $\mathbf{y}$ di dimensione $n \leq 60$. Scrivere un programma FORTRAN che

1. legge la dimensione n, i vettori $\mathbf{x}$ e $\mathbf{y}$; stampa i dati letti con commento;
2. calcola il prodotto scalare $\beta = \mathbf{x} \cdot \mathbf{y}$ servendosi della function SCAL, stampa il valore β;
3. dopo aver costruito il vettore $\mathbf{z} = \mathbf{x} + \beta\mathbf{y}$ calcola la norma assoluta di $\mathbf{z}$ impiegando la function NASS, stampa la norma trovata.