

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

Tema d'esame: soluzione di un particolare sistema di equazioni *integrali* utilizzando il metodo di Newton multidimensionale.

Descrizione del metodo di calcolo

Consideriamo le seguenti due funzioni $F : \mathbf{R}^2 \mapsto \mathbf{R}$ e $G : \mathbf{R}^2 \mapsto \mathbf{R}$, ciascuna delle quali è definita a partire da un integrale:

$$(1) \quad F(x, y) = \int_0^x f(t, y) dt, \quad G(x, y) = \int_0^y g(t, x) dt,$$

dove

$$(2) \quad f(t, y) = \exp\left(1 + \frac{\sin^2(t^2 + y^2)}{10}\right), \quad g(t, x) = \operatorname{artg}\left(10 + \exp\left(-\frac{t^2 x^2}{10^6}\right)\right).$$

Si noti che nella formula (1), la definizione di F coinvolge un *integrale a una sola variabile* (cioè t), il quale è tale che in esso la y può essere vista come un parametro; ovviamente, altrettanto si può dire della definizione di G dove la x gioca il ruolo del parametro.

Vogliamo studiare il seguente sistema di due equazioni *nonlineari* nelle due incognite ξ_1, ξ_2 :

$$(3) \quad \begin{cases} F(\xi_1, \xi_2) - c_F = 0 \\ G(\xi_1, \xi_2) - c_G = 0 \end{cases},$$

dove c_F e c_G sono due costanti reali i cui valori sono da considerarsi noti.

Per poter descrivere efficacemente il metodo di soluzione del precedente sistema di equazioni, è conveniente riscriverlo utilizzando una notazione vettoriale, cioè

$$(4) \quad \underline{\Phi}(\underline{\xi}) - \underline{c} = \underline{0},$$

dove la funzione $\underline{\Phi} : \mathbf{R}^2 \mapsto \mathbf{R}^2$ (che quindi è un *campo vettoriale*) è costituita dagli integrali che compaiono nei membri di sinistra delle equazioni che compongono il sistema (3) (cioè $\Phi_1(\xi_1, \xi_2) = F(\xi_1, \xi_2)$, $\Phi_2(\xi_1, \xi_2) = G(\xi_1, \xi_2)$); inoltre, ovviamente, sono stati utilizzati i seguenti simboli vettoriali $\underline{\xi} = (\xi_1, \xi_2)$ e $\underline{c} = (c_F, c_G)$.

I metodi numerici "alla Newton" sono innescati a partire da una approssimazione iniziale "buona abbastanza" della soluzione. Nel caso del problema che stiamo considerando, chiamiamo $\underline{\xi}_0$ tale approssimazione, che può essere determinata così come verrà spiegato in seguito. L'applicazione del metodo di Newton all'equazione (4) consiste nella costruzione di una successione di vettori bidimensionali $\{\underline{\xi}_n\}_{n \geq 0}$ che converge proprio alla soluzione del sistema (3). La definizione degli elementi della successione $\{\underline{\xi}_n\}_{n \geq 0}$ è per induzione, ovvero, a partire da $\underline{\xi}_0$ si determina il generico elemento $\underline{\xi}_{n+1} = \underline{\xi}_n - \underline{d}\underline{\xi}_n \forall n \geq 0$, con $\underline{d}\underline{\xi}_n \in \mathbf{R}^2$ soluzione del seguente sistema lineare

$$(5) \quad \mathcal{J}(\underline{\xi}_n) \underline{d}\underline{\xi}_n = \underline{\Phi}(\underline{\xi}_n) - \underline{c},$$

dove $\mathcal{J}(\underline{\xi})$ è la matrice Jacobiana 2×2 -dimensionale associata alla funzione $\underline{\Phi}$ (cioè la sua generica componente i, j -esima è tale che $\mathcal{J}_{i,j} = \partial\Phi_i/\partial\xi_j$), quindi essa è data dalla seguente formula:

$$(6) \quad \mathcal{J}(\underline{\xi}) = \begin{pmatrix} f(\xi_1, \xi_2) & \int_0^{\xi_1} \frac{\xi_2 \sin [2(t^2 + \xi_2^2)] \exp \left(1 + \frac{\sin^2(t^2 + \xi_2^2)}{10}\right)}{5} dt \\ \int_0^{\xi_2} \frac{-\frac{2t^2\xi_1}{10^6} \exp \left(-\frac{t^2\xi_1^2}{10^6}\right)}{1 + \left[10 + \exp \left(-\frac{t^2\xi_1^2}{10^6}\right)\right]^2} dt & g(\xi_2, \xi_1) \end{pmatrix}.$$

Questa è una trattazione troppo limitata per discutere le condizioni generali per le quali il metodo di Newton multidimensionale funziona efficacemente; nel caso del problema considerato, basti quindi sapere che *se* la prima approssimazione $\underline{\xi}_0$ è *abbastanza vicina* alla soluzione, allora *si può dimostrare* che la successione $\{\underline{\xi}_n\}_{n \geq 0}$ è convergente e, detto $\underline{\xi}_\infty = \lim_{n \rightarrow +\infty} \underline{\xi}_n$, sussiste l'equazione

$$\underline{\Phi}(\underline{\xi}_\infty) - \underline{c} = \underline{0},$$

in altri termini, il limite $\underline{\xi}_\infty$ della suddetta successione è *proprio soluzione* dell'equazione (4) e, quindi, anche del sistema (3).

Obiettivo (intermedio) 1:

si scriva un programma in linguaggio **C** che calcola numericamente il primo integrale definito che compare nella formula (1). Il programma deve contenere:

- (A) una *function* con due argomenti, che sono le variabili reali t e y ; (*alla fine della chiamata*) tale *function* deve restituire il valore di $f(t, y)$, così come è definita in (2);
- (B) una *function* che ha 5 argomenti: gli estremi a e b di un intervallo di integrazione, il numero di sotto-intervalli *numsubint*, un parametro reale β e infine un *puntatore a una function* φ , la quale, a sua volta, dipenderà da altri due argomenti (che, per il momento, indichiamo con i due simboli τ e β che sono di tipo **double**); *alla fine della chiamata*, tale *function* deve restituire il valore approssimato dell'integrale definito $\int_a^b \varphi(\tau, \beta) d\tau$, che viene calcolato utilizzando il *metodo del punto medio* basato su una griglia di *numsubint* sotto-intervalli;
- (C) l'*input da tastiera* di un paio di fissati valori delle variabili reali x e y (ciò può essere effettuato all'interno della *main function*); ciascuno di questi numeri deve essere sottoposto a un test in modo tale che, se x non è compreso nell'intervallo $[-50, 50]$, esso deve essere *reinserito correttamente*; altrettanto si faccia a proposito di y ;
- (D) un'*opportuna chiamata della function descritta al punto (B)*, in modo da calcolare numericamente l'integrale $F(x, y) = \int_0^x f(t, y) dt$; il valore di tale integrale deve essere approssimato utilizzando $NUMINTERV = 50\,000$ sotto-intervalli dell'insieme di integrazione $[0, x]$; inoltre, la suddetta *chiamata della function relativa al punto (B)*

deve essere tale che, tra gli argomenti, viene passato anche l'indirizzo della *function* descritta al punto (A);

- (E) la stampa sul video del valore approssimato dell'integrale $F(x, y) = \int_0^x f(t, y) dt$, così come è stato ottenuto grazie al procedimento descritto al precedente punto (D);

Alcuni consigli

È sicuramente utile (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).

Può essere comodo stabilire il valore di ciascuno dei parametri interi *NUMINTERV*, *N* e *NDIM* (questi ultimi due verranno introdotti in seguito) per mezzo di tre opportune direttive `#define`.

Obiettivo (intermedio) 2:

si integri il programma richiesto dall'obiettivo 1, in modo tale da effettuare il calcolo numerico del secondo integrale definito che compare nella formula (1). A tal fine si può procedere come segue:

- (A) si scriva una *function* con due argomenti, che sono le variabili reali t e x ; (*alla fine della chiamata*) tale *function* deve restituire il valore di $g(t, x)$, così come è definita in (2);
- (B) si effettui un'opportuna chiamata della *function* descritta al punto (B) dell'obiettivo 1, in modo da calcolare numericamente l'integrale $G(x, y) = \int_0^y g(t, x) dt$; il valore di tale integrale deve essere approssimato utilizzando $NUMINTERV = 50\,000$ sotto-intervalli dell'insieme di integrazione $[0, y]$; inoltre, la suddetta *chiamata della function relativa al punto (B) dell'obiettivo 1* deve essere tale che, tra gli argomenti, viene passato anche l'indirizzo della *function* descritta al punto (A);
- (C) all'interno della *main function*, si effettui la stampa sul video del valore approssimato dell'integrale $G(x, y) = \int_0^y g(t, x) dt$, così come è stato ottenuto grazie al procedimento descritto al precedente punto (B);

Obiettivo (intermedio) 3:

si integri il programma richiesto dall'obiettivo 2, in modo tale da aggiungere il calcolo di un'opportuna approssimazione iniziale della soluzione del sistema (3). A tale scopo si proceda come segue:

- (A) all'interno della *main function*, si effettui l'*input da tastiera* di un paio di fissati valori dei parametri c_F e c_G ; ciascuno di questi numeri deve essere sottoposto a un test in modo tale che, se c_F non è compreso nell'intervallo $[0, 100]$, esso deve essere *reinserto correttamente*; altrettanto si faccia a proposito di c_G ;

(B) si introduca il tipo di dato `fvett`, che è una `struct` definita dalle seguenti istruzioni:

```
struct campovett {
    double x;
    double y;
    double Fxy;
    double Gxy;
};
typedef struct campovett fvett;
```

inoltre, si definisca un *array* \mathcal{C} che è $(N+1) \times (N+1)$ -dimensionale e contiene elementi che sono del tipo `fvett`; si ponga $N = 8$ grazie a un'opportuna direttiva `#define`;

- (C) si determinino gli elementi dell'*array* \mathcal{C} in modo tale che contengano i valori del *campo vettoriale* $\Phi : \mathbf{R}^2 \mapsto \mathbf{R}^2$ corrispondenti a una "griglia" regolare di punti, così come descritto ai seguenti punti (C1)–(C3);
- (C1) si iterino le istruzioni descritte ai seguenti punti (C2)–(C3), quando gli indici interi i e j "corrono" (indipendentemente l'uno dall'altro) tra 0 e N ;
- (C2) si pongano i valori di x e y dell' i, j -esimo elemento dell'*array* \mathcal{C} in modo tale che siano, rispettivamente, uguali a $i \cdot c_F / (N \cdot e)$ e $j \cdot c_G / [N \cdot \text{artg}(10)]$;
- (C3) si pongano i valori di Fxy e Gxy dell' i, j -esimo elemento dell'*array* \mathcal{C} in modo tale che siano, rispettivamente, uguali agli integrali $F(x, y)$ e $G(x, y)$ che sono definiti nella formula (1) e sono da calcolare come descritto negli obiettivi 1 e 2, ma in corrispondenza ai valori di x e y introdotti al precedente punto (C2);
- (D) si selezioni tra gli elementi dell'*array* \mathcal{C} quello che costituisce la migliore approssimazione della soluzione del sistema (3) tra quelli appartenenti alla "griglia" descritta ai precedenti punti (C)–(C3); tale migliore approssimazione viene individuata così come descritto ai seguenti punti (D1)–(D2);
- (D1) si considerino solo quegli elementi dell'*array* \mathcal{C} tali che i corrispondenti valori di Fxy e Gxy sono rispettivamente $\geq c_F$ e $\leq c_G$;
- (D2) tra gli elementi dell'*array* \mathcal{C} che soddisfano la caratteristica descritta in (D1) si selezioni quello tale che la distanza $\sqrt{(Fxy - c_F)^2 + (Gxy - c_G)^2}$ è minima;
- (E) siano \bar{x} , \bar{y} , \bar{F} e \bar{G} i valori delle variabili x , y , Fxy e Gxy che sono alloggiare all'interno dell'elemento dell'*array* \mathcal{C} che costituisce la migliore approssimazione, la quale è stata determinata così come descritto ai precedenti punti (D)–(D2); si *stampino ordinatamente* sul video i suddetti valori di \bar{x} , \bar{y} , \bar{F} e \bar{G} ;

Obiettivo (finale) 4:

si integri il programma richiesto dall'obiettivo 3, in modo tale da effettuare il calcolo della soluzione del sistema (3), applicando in modo opportuno il metodo di Newton multidimensionale. A tal fine si proceda come segue:

- (A) all'interno della *main function*, si definiscano i vettori \underline{c} e $\underline{\xi}_0$, in modo tale che $\underline{c} = (c_F, c_G)$ e $\underline{\xi}_0 = (\bar{x}, \bar{y})$; ciò è utile perché le *function* che verranno descritte in seguito hanno sempre degli *array* come argomenti;
- (B) si scriva una *function* che ha tre argomenti: i vettori $\underline{\xi}$, \underline{c} e $\underline{\eta}$; *alla fine della chiamata*, il

vettore $\underline{\eta}$ dovrà essere uguale a $\underline{\Phi}(\underline{\xi}) - \underline{c}$ (cioè $\eta_1 = F(\xi_1, \xi_2) - c_F$, $\eta_2 = G(\xi_1, \xi_2) - c_G$, dove F e G sono definiti nella formula (1));

- (C) si scriva una *function* che ha due argomenti: il vettore $\underline{\xi}$, e la matrice \mathcal{J} ; *alla fine della chiamata*, tale matrice sarà tale che $\mathcal{J} = \mathcal{J}(\underline{\xi})$, così come essa è definita nell'equazione (6); si noti che per effettuare un tale calcolo, si devono utilizzare le *function* descritte ai punti (A) e (B) dell'obiettivo 1 e al punto (A) dell'obiettivo 2, inoltre, si devono scrivere altre due *function* che restituiscono il valore delle integrande che compaiono in (6);
- (D) si includano tutte le *function* necessarie per il calcolo della soluzione di un sistema lineare del tipo $A\underline{\xi} = \underline{b}$, dove \underline{b} e $\underline{\xi}$ sono vettori $NDIM$ -dimensionali e A una matrice $NDIM \times NDIM$ -dimensionale; sia il valore del parametro intero $NDIM$ posto uguale a 2 grazie a un'opportuna direttiva `#define`;
- (E) si scriva una *function* che ha come argomento un *vettore* $NDIM$ -dimensionale \underline{v} e restituisce la sua *norma euclidea* $\|\underline{v}\|$, che è definita come segue:

$$\|\underline{v}\| = \sqrt{\sum_{j=1}^{NDIM} v_j^2};$$

- (F) si scriva una *function* che ha tre argomenti: i vettori $\underline{\xi}_0$, \underline{c} e $\underline{\xi}$, dove $\underline{\xi}_0$ ha il significato di “approssimazione iniziale”, mentre (*alla fine dell'esecuzione di tale function*) $\underline{\xi}$ conterrà i valori della soluzione del sistema (3) (in seguito all'applicazione del metodo di Newton); questa *function* traduce in linguaggio di programmazione la procedura descritta ai seguenti punti (a)–(d);
- (a) si ponga $\underline{\xi} = \underline{\xi}_0$;
- (b) si iterino le istruzioni descritte ai seguenti punti (c1)–(c5) in modo tale da costituire un ciclo, la condizione di permanenza all'interno del quale è descritta al punto (d);
- (c1) si calcoli il “vettore noto” $\underline{b} = \underline{\Phi}(\underline{\xi}) - \underline{c}$ dell'equazione (5) grazie a una chiamata alla *function* richiesta al precedente punto (B);
- (c2) si calcoli la matrice $A = \mathcal{J}(\underline{\xi})$ dell'equazione (5) grazie a una chiamata alla *function* richiesta al punto (C);
- (c3) si calcoli la soluzione $\underline{d\xi}$ dell'equazione lineare $A\underline{d\xi} = \underline{b}$, chiamando in modo opportuno una delle *function* richieste al punto (D);
- (c4) si aggiorni il valore della soluzione (*approssimata*) $\underline{\xi}$ ponendolo uguale a $\underline{\xi} - \underline{d\xi}$;
- (c5) si ponga la quantità $\sigma = \|\underline{d\xi}\|/\|\underline{\xi}\|$, dove le *norme euclidee* sono calcolate grazie a delle opportune chiamate della *function* descritta al precedente punto (E);
- (d) se $\sigma > 10^{-10}$ si ripeta l'esecuzione delle istruzioni descritte ai precedenti punti (c1)–(c5);
- (G) si modifichi la *main function* in modo tale che, a sua volta, essa contenga anche:
- (G1) la chiamata della *function* descritta al precedente punto (F), in modo da effettuare il calcolo della soluzione del sistema (3);
- (G2) la stampa ordinata sul video di ciascuna delle componenti della soluzione $\underline{\xi}$; inoltre, si stampino *in formato esponenziale* le componenti del corrispondente vettore $\underline{\eta} = \underline{\Phi}(\underline{\xi}) - \underline{c}$; ovviamente, la verifica della soluzione del sistema (3) è da

considerarsi ben riuscita se entrambe le componenti di $\underline{\eta}$ sono dello stesso ordine di grandezza dell'*errore di macchina* o, alla peggio, sono poco più grandi.