

Prova d'esame di Laboratorio di Calcolo I
per il corso di laurea in Matematica
15 Febbraio 2011

Tema d'esame: studio del sistema di *equazioni differenziali* di Lotka–Volterra.

Descrizione del metodo di calcolo

Le equazioni di Lotka–Volterra sono ben note perché forniscono un modello matematico in grado di descrivere la dinamica di un ecosistema in cui interagiscono soltanto due specie animali: una delle due come predatore, l'altra come la sua preda. Questa modellizzazione matematica è stata proposta indipendentemente da Alfred J. Lotka nel 1925 e Vito Volterra nel 1926 ed è riassunta dal seguente sistema di equazioni differenziali:

$$(1) \quad \begin{cases} \dot{x}_1 = \alpha_1 x_1 - \alpha_2 x_1 x_2 \\ \dot{x}_2 = -\alpha_3 x_2 + \alpha_4 x_1 x_2 \end{cases},$$

dove le incognite sono le due funzioni $x_1 = x_1(t)$, $x_2 = x_2(t)$ e $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ sono quattro parametri *positivi* i cui valori sono da considerarsi fissati. Le due funzioni $x_1 = x_1(t)$, $x_2 = x_2(t)$ descrivono l'andamento, nel tempo, delle quantità di prede e di predatori, rispettivamente. Il problema di Cauchy associato al modello è formato dalle equazioni che compaiono in (1) e dalle seguenti condizioni iniziali:

$$(2) \quad \begin{cases} x_1(0) = \bar{x}_1 \\ x_2(0) = \bar{x}_2 \end{cases}.$$

Se almeno uno dei due valori \bar{x}_1, \bar{x}_2 è nullo, allora si verifica facilmente che la corrispondente funzione (cioè $x_1(t)$ oppure $x_2(t)$) è identicamente nulla e l'altra si risolve separatamente, senza che vi sia quindi interazione tra le due specie, ovvero le prede e i predatori. Il caso interessante è quindi quando sia \bar{x}_1 che \bar{x}_2 sono positivi; conseguentemente $x_1(t) > 0$ e $x_2(t) > 0, \forall t \in \mathbf{R}$.

Sia la funzione $\underline{f} : \mathbf{R}^2 \mapsto \mathbf{R}^2$ il *campo vettoriale* associato a un sistema generale di equazioni differenziali del primo ordine, cioè

$$(3) \quad \underline{\dot{x}} = \underline{f},$$

dove $\underline{f} = \underline{f}(\underline{x})$. Ad esempio, nel caso del sistema (1) si ha che

$$(4) \quad \underline{f}(\underline{x}) = \begin{pmatrix} \alpha_1 x_1 - \alpha_2 x_1 x_2 \\ -\alpha_3 x_2 + \alpha_4 x_1 x_2 \end{pmatrix}.$$

Ricordiamo che se il campo vettoriale $\underline{f} : \mathbf{R}^2 \mapsto \mathbf{R}^2$ è una funzione regolare lo sarà anche la soluzione $t \mapsto \underline{x}(t)$ del problema di Cauchy associato. In particolare, nel caso del problema di Cauchy formato dalle equazioni (1) e (2) sia $\underline{f} : \mathbf{R}^2 \mapsto \mathbf{R}^2$ che $\underline{x} : \mathbf{R} \mapsto \mathbf{R}^2$ sono funzioni \mathcal{C}^∞ .

I metodi di tipo Runge–Kutta sono noti per essere i più semplici tra quelli numerici che permettono di approssimare la soluzione di un sistema di equazioni differenziali in forma esplicita. In particolare, descriviamo qui di seguito il metodo di Heun. Supponiamo di conoscere la soluzione $\underline{x}(\tau)$ a un fissato istante τ (o almeno una sua approssimazione)

e ci proponiamo di calcolarla anche al tempo $\tau + h$ (dove si intende che il valore di h è piccolo in valore assoluto). A tal fine, determiniamo i vettori \underline{k}_1 , \underline{k}_2 e \underline{x}^* in modo tale che

$$(5) \quad \underline{k}_1 = \underline{f}(\underline{x}(\tau)) , \quad \underline{x}^* = \underline{x}(\tau) + h\underline{k}_1 , \quad \underline{k}_2 = \underline{f}(\underline{x}^*) .$$

Sussiste quindi la seguente relazione:

$$(6) \quad \underline{x}(\tau + h) \simeq \underline{x}(\tau) + h \frac{\underline{k}_1 + \underline{k}_2}{2} .$$

Più esattamente, supponiamo di essere interessati a calcolare numericamente la soluzione $t \mapsto \underline{x}(t)$ per un intervallo di tempo $[0, T]$, che suddividiamo in N sottointervalli $[\tau_{j-1}, \tau_j]$ di ampiezza $h = T/N$, in modo da utilizzare il metodo di Heun per approssimare $\underline{x}(\tau_j) \forall j = 1, \dots, N$. Si dimostra che alla fine del nostro procedimento, il calcolo di $\underline{x}(T)$ sarà corretto a meno di un errore $\mathcal{O}(h^2)$. In questo senso il metodo di Heun è di tipo Runge–Kutta di ordine 2.

Le equazioni (1) di Lotka–Volterra hanno alcune caratteristiche particolari. Infatti, esiste un solo punto \underline{X} di equilibrio all'interno del quadrante $x_1 > 0$, $x_2 > 0$; esso ha coordinate:

$$(7) \quad \underline{X} = \left(\frac{\alpha_3}{\alpha_4}, \frac{\alpha_1}{\alpha_2} \right) .$$

Inoltre, esiste una funzione $F : \mathbf{R}_+ \times \mathbf{R}_+ \mapsto \mathbf{R}$ che è una costante del moto, cioè $\forall t \in \mathbf{R}$ $F(\underline{x}(t)) = F(\underline{x}(0))$, essa è definita come segue:

$$(8) \quad F(\underline{x}) = -\alpha_3 \log x_1 + \alpha_4 x_1 - \alpha_1 \log x_2 + \alpha_2 x_2 .$$

Essendo la funzione F evidentemente costituita da due parti *separabili*, l'una che dipende solo da x_1 , e l'altra solo da x_2 , si verifica facilmente che F ha uno e un solo punto stazionario in corrispondenza proprio al *punto di minimo* \underline{X} , le cui coordinate sono riportate nella definizione (7).

Obiettivo (intermedio) 1:

si scriva un programma in linguaggio **C** che verifica numericamente che il punto \underline{X} , definito in (7), è di equilibrio per il sistema di equazioni differenziali di Lotka–Volterra. Il programma deve contenere:

- (A) una *function* con tre argomenti, che sono i vettori di variabili reali \underline{x} , $\underline{\alpha}$ e \underline{y} ; il vettore $\underline{x} \in \mathbf{R}^2$ e quello dei parametri $\underline{\alpha} \in \mathbf{R}^4$, sono da intendersi come le variabili di *input*; la *function* deve essere scritta in modo tale che (*alla fine della chiamata*) i valori delle componenti del vettore \underline{y} soddisfano l'equazione $\underline{y} = \underline{f}(\underline{x})$, dove il campo vettoriale $\underline{f} = \underline{f}(\underline{x})$ è definito in (4);
- (B) l'*input da tastiera* dei valori fissati delle componenti del vettore di parametri $\underline{\alpha}$ (ciò può essere effettuato all'interno della *main function*); ciascuna delle componenti α_i deve essere sottoposta a un test in modo tale che, se α_i non è *positiva*, essa deve essere *reinserta correttamente*;
- (C) il calcolo (banale) delle componenti del punto di equilibrio \underline{X} , così come sono definite dalla formula (7);
- (D) un'*opportuna chiamata della function descritta al punto (A)*, in modo da calcolare le componenti del campo vettoriale in corrispondenza al punto di equilibrio, cioè $\underline{f}(\underline{X})$;

- (E) la stampa sul video *in formato esponenziale* dei valori delle componenti di $\underline{f}(\underline{X})$; ovviamente, la verifica del fatto che \underline{X} è un punto di equilibrio del sistema (1) è da considerarsi ben riuscita, se entrambe le componenti di $\underline{f}(\underline{X})$ sono dello stesso ordine di grandezza dell'*errore di macchina* o, alla peggio, sono poco più grandi.

Un primo consiglio

Può essere comodo stabilire che il valore di due parametri interi $NDIM$ e $NPARAM$ sono rispettivamente uguali a 2 e a 4, per mezzo di due opportune direttive `#define`. Sarà così possibile porre uguale a $NDIM$ il numero di componenti di tutti i vettori che hanno le stesse dimensioni del campo vettoriale, mentre il numero di componenti del vettore α può essere posto uguale a $NPARAM$. Con questo approccio, si può scrivere il programma in una forma che riduce al minimo le modifiche da apportare per considerare un modello esteso a un numero di dimensioni > 2 .

Obiettivo (intermedio) 2:

si integri il programma richiesto dall'obiettivo 1, in modo tale da effettuare una verifica numerica del fatto che il punto \underline{X} è di minimo per la costante del moto F definita in (8). A tal fine (si finga di ignorare che tale verifica è banale da farsi con un procedimento analitico e) si proceda come segue:

- (A) si scriva una *function* con due argomenti, che sono il vettore $\underline{x} \in \mathbf{R}^2$ e quello dei parametri $\underline{\alpha} \in \mathbf{R}^4$; la *function* deve essere scritta in modo tale che (*alla fine della chiamata*), essa restituisca il valore assunto dalla costante del moto in corrispondenza a \underline{x} (e a $\underline{\alpha}$), cioè $F(\underline{x})$, definito come in formula (8);
- (B) si effettui *un'opportuna chiamata della function descritta al punto (A)*, in modo da calcolare $F^* = F(\underline{X})$;
- (C) all'interno della *main function*, si effettui *l'input da tastiera* del valore di una variabile m , cui daremo il significato di coefficiente angolare di una retta passante per \underline{X} ;
- (D) si calcoli il valore di $h^* = \min\{X_1, X_2/m\}$;
- (E) si iteri l'esecuzione delle istruzioni descritte ai seguenti punti (E1)–(E4), quando un *contatore* intero j va da 1 a 99;
- (E1) si ponga $h = (jh^*)/100$;
- (E2) si ponga $x_1 = X_1 + h$ e $x_2 = X_2 + mh$, cioè \underline{x} appartiene alla retta passante per \underline{X} e di coefficiente angolare m ; inoltre \underline{x} è il punto corrispondente all'ascissa $X_1 + h$ (si osservi che i valore di h e h^* sono scelti in modo tale che entrambe le componenti di \underline{x} sono positive);
- (E3) si effettui *un'opportuna chiamata della function descritta al punto (A)*, in modo da calcolare $F_{\underline{x}} = F(\underline{x})$;
- (E4) *se succede che $F_{\underline{x}} \leq F^*$, allora si scriva sul video un messaggio che evidenzia l'anomalia (perché dovrebbe essere sempre verificato che $F_{\underline{x}} > F^*$) e si arresti immediatamente l'esecuzione del programma;*
- (F) si iteri l'esecuzione delle istruzioni descritte ai suddetti punti (E1)–(E4), *anche* quando un *contatore* intero j va da -1 a -99 ;
- (G) si scriva sul video un messaggio di conferma che il punto \underline{X} è di minimo per la costante del moto F (si osservi che, se il programma arriva ad eseguire la presente prescrizione (G), significa che non ha mai eseguito la parte del punto (E4) dove si richiedeva l'arresto immediato del programma stesso);

Obiettivo (intermedio) 3:

si integri il programma richiesto dall'obiettivo 2, in modo tale da aggiungere il calcolo numerico della soluzione del sistema (1) e la verifica che il metodo di Heun è di ordine 2. A tale scopo si proceda come segue:

- (A) si scriva una *function* con quattro argomenti, che sono l'intervallo di tempo h , il vettore $\underline{x} \in \mathbf{R}^2$, quello dei parametri $\underline{\alpha} \in \mathbf{R}^4$ e infine un *puntatore a una function* f , la quale effettua il calcolo del campo vettoriale e, a sua volta, dipenderà da altri tre argomenti (che sono tre array di tipo `double`); la *function* deve essere scritta in modo tale che, *se all'inizio della chiamata* supponiamo che nel vettore \underline{x} siano memorizzati i valori corrispondenti alla soluzione $\underline{x}(\tau)$ a un certo istante τ , *allora, alla fine della chiamata*, nello stesso vettore \underline{x} saranno presenti i valori corrispondenti all'approssimazione di $\underline{x}(\tau + h)$ che è definita nel membro di destra della formula (6), dove i valori dei vettori \underline{k}_1 , \underline{k}_2 e \underline{x}^* sono dati in (5);
- (B) all'interno della *main function*, si effettui l'*input da tastiera* delle condizioni iniziali $\underline{x}(0) = (\bar{x}_1, \bar{x}_2)$; ciascuna delle componenti \bar{x}_i (per $i = 1, 2$) deve essere sottoposta a un test in modo tale che, se \bar{x}_i non è positiva, essa deve essere *reinserita correttamente*;
- (C) all'interno della *main function*, si effettui l'*input da tastiera* dell'istante di tempo T al quale arrestiamo il calcolo numerico della soluzione delle equazioni differenziali; il valore di T deve essere sottoposto a un test in modo tale che, se T è nullo, esso deve essere *reinserito correttamente*;
- (D) si effettui un'opportuna chiamata della *function* descritta al punto (A) dell'obiettivo 2, in modo da calcolare $F_0 = F(\underline{x}(0))$;
- (E) si iteri l'esecuzione delle istruzioni descritte ai seguenti punti (E1)–(E5), quando un contatore intero i va da *IMIN* a *IMAX*;
 - (E1) si ponga $N = 10^i$ e $h = T/N$;
 - (E2) si ponga il vettore \underline{x} uguale alle condizioni iniziali, in modo tale che $x_1 = \bar{x}_1$ e $x_2 = \bar{x}_2$;
 - (E3) si effettuino N chiamate consecutive della *function* descritta al punto (A); inoltre, le suddette chiamate devono essere tali che, tra gli argomenti, viene passato anche l'indirizzo della *function* descritta al punto (A) dell'obiettivo 1; si osservi che l'effetto di ciascuna di queste N chiamate consecutive è quello di produrre il calcolo approssimato di $\underline{x}(\tau_j)$ a partire da $\underline{x}(\tau_{j-1})$, dove $\tau_j = jh \forall j = 0, \dots, N$;
 - (E4) si calcoli $F_T = F(\underline{x}(T))$, effettuando un'opportuna chiamata della *function* descritta al punto (A) dell'obiettivo 2; qui si deve intendere che $\underline{x}(T)$ non è la soluzione esatta al tempo T (che peraltro è ignota), ma la sua approssimazione calcolata come spiegato al precedente punto (E3);
 - (E5) si stampi sul video il valore attuale di N e in formato esponenziale il corrispondente errore $|F_T - F_0|$.

Alcuni altri 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).

Al fine di far funzionare il programma in modo da effettuare dei test che siano contemporaneamente significativi e che terminino in tempi ragionevoli, è bene stabilire che il

valore dei due parametri interi $IMIN$ e $IMAX$ sono rispettivamente uguali a 3 e a 7, per mezzo di due opportune direttive `#define`. Sarà così possibile modificare velocemente e in modo coerente il programma, a seconda delle proprie esigenze e della velocità della CPU a disposizione.

Ovviamente, la verifica del fatto che il metodo di Heun è di ordine 2 deve considerarsi ben riuscita, se si osserva che *ogni qualvolta che il numero di sottointervalli N viene aumentato di un fattore 10, l'errore $|F_T - F_0|$ diminuisce circa di un fattore 100*.

Obiettivo (finale) 4:

al programma richiesto dagli obiettivi precedenti si aggiunga la scrittura ordinata su **file** di una successione di coppie di valori $(x_1(\tau_j), x_2(\tau_j)) \forall j = 1, \dots, 10000$, a partire dalla quale è possibile tracciare il grafico di un'orbita associata alla soluzione del problema di Cauchy descritto dalle formule (1)–(2). Inoltre, si compiano le operazioni necessarie al fine di visualizzare tale grafico, utilizzando il software **gnuplot**. A tali scopi si può procedere come segue:

- (A) si apra un **file** che è destinato a contenere le coppie dei punti della soluzione approssimata delle equazioni differenziali;
- (B) si ponga $i = 4$ (ciò corrisponde a imporre che il numero di sottointervalli N sia uguale a 10 000, si veda il seguito del testo);
- (C) si eseguano ancora le istruzioni descritte ai precedenti punti (E1)–(E4) dell'obiettivo 3, ma facendo attenzione alla modifica richiesta al punto seguente;
- (D) si modifichi il punto (E4) dell'obiettivo 3, in modo tale che all'interno del ciclo che è implicitamente richiesto proprio dal punto (E4) si proceda alla stampa su **file**; tale stampa deve essere effettuata di modo che sulla j -esima riga del **file** compaia prima il valore di $x_1(\tau_j)$ e poi quello corrispondente di $x_2(\tau_j)$;
- (E) si chiuda il **file** che era stato aperto al punto (A);
- (F) *separatamente dal programma*, si scriva un nuovo **file** contenente i comandi necessari al software **gnuplot**, al fine di far apparire sullo schermo un grafico (esteticamente apprezzabile) dell'orbita associata alla soluzione del problema di Cauchy, che è stata memorizzata su **file** così come è stato descritto ai precedenti punti (A)–(E);