Prova d'esame di Laboratorio di Calcolo I
per il corso di laurea in Matematica
24 Febbraio 2014

Tema d'esame: studio di alcune proprietà dei numeri di Bernoulli.

Descrizione del metodo di calcolo

I numeri di Bernoulli $\mathcal{B}_m$ costituiscono una successione di numeri razionali che permettono di risolvere alcuni problemi in vari campi della matematica. Esistono parecchi modi equivalenti di definire i numeri di Bernoulli, qui di seguito introduciamo preliminarmente un'opportuna generalizzazione. Siano i polinomi (di Bernoulli) $B_m : \mathbf{R} \to \mathbf{R}$ tali che

$$B_m(x) = x^m - \sum_{k=0}^{m-1} \left[\binom{m}{k} \frac{B_k(x)}{m-k+1} \right] \quad \forall\, m \geq 1 \,, \qquad B_0(x) = 1 \,. \tag{1}$$

Nella formula precedente, che è evidentemente di tipo *ricorsivo*, compare il coefficiente binomiale:

$$\binom{m}{k} = \frac{m!}{(m-k)!\,k!} \,. \tag{2}$$

La successione dei numeri di Bernoulli $\{\mathcal{B}_m\}_{m\geq 0}$ è tale che

$$\mathcal{B}_m = B_m(0) \quad \forall\, m \geq 0 \,. \tag{3}$$

Si possono dimostrare le seguenti proprietà elementari:

(i) quando l'indice m assume un valore che è contemporaneamente dispari e maggiore o uguale a 3, allora $\mathcal{B}_m = 0$;

(ii) gli elementi della successione, che sono diversi da zero, hanno segni alterni; ad esempio, $\mathcal{B}_0 = 1$, $\mathcal{B}_1 = -1/2$, $\mathcal{B}_2 = 1/6$ $\mathcal{B}_4 = -1/30$, $\mathcal{B}_6 = 1/42$, etc.;

(iii) l'uguaglianza $B_m(1) = \mathcal{B}_m$ vale per ogni $m \neq 1$; invece, per $m = 1$ si ha che $B_1(1) = -\mathcal{B}_1 = 1/2$.

I numeri di Bernoulli, tra le altre, soddisfano anche la seguente equazione:

$$\mathcal{B}_m = (-1)^{m/2}\, 2 \int_0^\infty \frac{m\, x^{m-1}}{1 - e^{2\pi x}}\, \mathrm{d}x \tag{4}$$

quando m è un numero *pari e positivo*. Inoltre, essi compaiono nelle espansioni in serie di Taylor della funzione tangente (trigonometrica) e della tangente iperbolica. Infatti, $\forall\, x \in \mathbf{R}$ tale che $|x| < \pi/2$, valgono le due seguenti equazioni:

$$\begin{aligned} \tan x &= \sum_{m=1}^{\infty} \left[\frac{\big(-1\big)^{m-1}\, 2^{2m}\big(2^{2m}-1\big)\mathcal{B}_{2m}}{(2m)!} x^{2m-1} \right] \,, \\ \tanh x &= \sum_{m=1}^{\infty} \left[\frac{2^{2m}\big(2^{2m}-1\big)\mathcal{B}_{2m}}{(2m)!} x^{2m-1} \right] \,. \end{aligned} \tag{5}$$

Esistono algoritmi di calcolo dei numeri di Bernoulli che sono assai più efficienti delle definizioni riassunte dalle formule (1)–(3). Ad esempio, quello proposto da Akiyama e Tanigawa[⋆] può essere riassunto come segue:

(α1) si definisca un array temporaneo $\mathcal{A}$ di componenti $\mathcal{A}_0$, $\mathcal{A}_1$, ..., $\mathcal{A}_m$, le quali a loro volta servono al calcolo di $\mathcal{B}_m$;

(α2) si iterino le operazioni descritte ai seguenti punti (α21)–(α22) mentre un contatore intero i va da 0 fino a m ;

- (α21) si ponga $\mathcal{A}_i = 1/(i+1)$;
- (α22) mentre un secondo contatore intero j va da i fino a 1 (ovviamente, procedendo a ritroso), si ridefinisca il valore di $\mathcal{A}_{j-1}$, in modo che sia $\mathcal{A}_{j-1} = j\big(\mathcal{A}_{j-1} - \mathcal{A}_j\big)$;

(α3) si restituisca il valore di $\mathcal{A}_0$ che (si può dimostrare che) è proprio uguale a quello di $\mathcal{B}_m$.

La successione dei numeri di Eulero $\{\mathcal{E}_m\}_{m\ge 0}$ è strettamente connessa a quella dei numeri di Bernoulli e, insieme ad essa, permette di definire degli approssimanti razionali di π . Infatti, si dimostra che

$$2 \lim_{m\to\infty} \left[\big(2^{2m} - 4^{2m}\big) \frac{\mathcal{B}_{2m}}{\mathcal{E}_{2m}} \right] = \pi \ , \tag{6}$$

dove

$$\mathcal{E}_m = \sum_{k=1}^{m} \left[\binom{m}{k-1} \frac{2^{2m} - 4^{2m}}{k} \mathcal{B}_k \right] \qquad \forall\, m \ge 0 \ . \tag{7}$$

Obiettivo (intermedio) 1:
si scriva un programma in linguaggio **C** che, per un dato valore dell'indice m , determina (approssimativamente) il corrispondente numero di Bernoulli $\mathcal{B}_m$, tramite il calcolo di un'opportuno integrale. Il programma deve contenere:

(A) una *function* che ha come argomenti la "variabile di integrazione" x e il parametro (intero) m ; *(alla fine della chiamata)* tale *function* deve restituire il valore di $m\,x^{m-1}/\big(1 - e^{2\pi x}\big)$, cioè proprio la funzione integranda che compare nel membro di destra dell'equazione (4);

(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 intero p e infine un *puntatore a una function f*, la quale, a sua volta, dipenderà da altri due argomenti di cui il primo è di tipo `double`, mentre il secondo è di tipo `int`; *(alla fine della chiamata)* tale *function*

[⋆] Sfortunatamente, l'algoritmo di Akiyama–Tanigawa effettua il calcolo dei numeri di Bernoulli, in modo tale che viene generato un rapidissimo accumulo degli errori. Siccome tutti i numeri di Bernoulli (e gli elementi del vettore $\mathcal{A}$) sono *razionali*, allora essi potrebbero essere calcolati senza alcun errore utilizzando opportune librerie di *function* che implementano l'*aritmetica in multi-precisione*. Un tale approccio (quando viene tradotto in un qualsiasi linguaggio di programmazione) ha lo svantaggio di richiedere un rapido aumento della quantità di memoria necessaria al calcolo di $\mathcal{B}_m$, al crescere dell'indice m . Ovviamente, la descrizione dei metodi dell'*aritmetica in multi-precisione* esula dagli scopi di questo testo.

deve restituire il valore approssimato dell'integrale definito $\int_a^b f(x,p)\,\mathrm{d}x$, che viene calcolato utilizzando il *metodo del punto medio* basato su una griglia di `numsubint` sotto-intervalli;

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

(C1) l'*input da tastiera* del valore dell'indice m ; nel programma, la variabile m deve essere di tipo `int`; il valore inserito in *input* deve essere sottoposto a un test in modo tale che, se *m non è un numero pari compreso tra* 2 *e* 24 , allora esso deve essere *reinserito correttamente*;

(C2) il calcolo del numero di Bernoulli $\mathcal{B}_m$, grazie alla valutazione del membro di destra dell'equazione (4), cioè la quantità $(-1)^{m/2}\,2\int_0^\infty \big[(m\,x^{m-1})/(1-e^{2\pi x})\big]\,\mathrm{d}x$, per mezzo di *un'opportuna chiamata della function descritta al punto (B)* e utilizzando il valore dell'indice m che è stato inserito in *input* così come richiesto al precedente punto (C1); il valore del suddetto integrale deve essere approssimato utilizzando 100 000 sotto-intervalli dell'insieme di integrazione $[0, 10\,m]$; inoltre, la *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);

(C3) la stampa sul video del valore del numero di Bernoulli $\mathcal{B}_m$, calcolato così come è stato richiesto al precedente punto (C2).

Alcuni consigli

È sicuramente comodo (e *prudente*) utilizzare (e, eventualmente, modificare) delle *function* 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 integri il programma richiesto dall'obiettivo 1, in modo tale da poter valutare il polinomio di Bernoulli $B_m(x)$, in corrispondenza a un argomento $x = n$ intero e utilizzando la formula ricorsiva (1). A tal fine, si proceda come segue:

(A) si scriva una *function* che ha due argomenti: gli indici m e k , entrambi di tipo `int`; *all'interno di questa nuova function, si sconsiglia* di tradurre la formula esplicita (2) in linguaggio di programmazione, perché genererebbe dei fattori intermedi così grandi che non potrebbero essere rappresentati esattamente sul calcolatore; per ovviare a questo problema, si consiglia invece di procedere come descritto ai seguenti punti (A1)–(A3):

(A1) si effettui la definizione iniziale di una *variabile locale* `ris` di tipo `double` in modo che sia `ris` $= 1$;

(A2) si iterino le operazioni descritte al seguente punto (A21), mentre un contatore intero j va da 1 fino a k ;

(A21) si aggiorni il valore di `ris` ponendolo uguale al risultato che si ottiene moltiplicando il "vecchio" valore di `ris` per $m - j + 1$, per poi dividere tale prodotto per j ;

(A3) si restituisca il valore di `ris` che, al termine del ciclo descritto ai precedenti punti (A2)–(A21), è proprio uguale al coefficiente binomiale definito in (2);

(B) si scriva un'altra *function* che ha due argomenti: l'indice m e la variabile n , entrambi di tipo `int`; tale *function deve essere di tipo ricorsivo* e al termine della chiamata deve restituire il valore di $B_m(n)$, che viene calcolato utilizzando la formula ricorsiva (1);

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

(C1) *l'input da tastiera* del valore dell'indice n; tale valore inserito in *input* deve essere sottoposto a un test in modo tale che, se n *non è compreso* tra 0 e 10, allora esso deve essere *reinserito correttamente*;

(C2) il calcolo della quantità $B_m(n)$ grazie a *un'opportuna chiamata della function descritta al punto (B)* e utilizzando i valori delle variabili intere m e n che sono stati inseriti così come richiesto al punto (C1) dell'obiettivo 1 e al precedente punto (C1), rispettivamente;

(C3) la stampa sul video del valore della quantità $B_m(n)$ appena calcolata così come è stato richiesto al precedente punto (C2).

Alcuni altri consigli

Porre $n = 0$ oppure $n = 1$ (nella fase di *input* richiesta al punto (C1)) è senz'altro un utile test per la correttezza del programma. Infatti, si ricordi che $\mathcal{B}_m = B_m(0) = B_m(1)$ $\forall\ m \geq 2$, quindi, quando $n = 0$ oppure $n = 1$, allora i valori stampati sullo schermo così come richiesto ad entrambi i punti (C3) (sia dell'obiettivo 1 che del presente obiettivo) devono essere uguali a meno degli inevitabili errori numerici.

Obiettivo (intermedio) 3:

si integri il programma richiesto dall'obiettivo 2, in modo tale da effettuare il calcolo "simultaneo" della funzione tangente (trigonometrica) e della tangente iperbolica, utilizzando le espansioni in serie che compaiono in formula (5). A tal fine, si può procedere come segue:

(A) si scriva una *function* che ha tre argomenti: il primo è un valore x di tipo `double`, mentre gli ultimi due *devono essere tali da restituire all'ambiente chiamante* i valori della tangente trigonometrica e della tangente iperbolica in corrispondenza a x (cioè, rispettivamente, le approssimazioni numeriche di $\tan(x)$ e $\tanh(x)$); qui di seguito indichiamo con `tang` e `tang_ip` le variabili che alla fine dell'esecuzione di tale *function* conterranno proprio i valori approssimati di $\tan(x)$ e $\tanh(x)$, rispettivamente; allo scopo di calcolarli, si proceda così come descritto ai seguenti punti (A1)–(A5);

(A1) si effettuino le definizioni iniziali di una coppia di *variabili locali* di tipo `int`, in modo che sia $m = 0$ e $\sigma = 1$;

(A2) si effettuino le definizioni iniziali di alcune *variabili locali* di tipo `double`, in modo che sia `fatt` $= 2$, `pot2` $= 4$, `potx` $= x$;

(A3) si ponga inizialmente `tang` $= 0$ e `tang_ip` $= 0$;

(A4) si iterino le operazioni descritte ai seguenti punti (A41)–(A49) mentre è verificata la condizione (di permanenza nel ciclo), che è espressa al punto (A5);

(A41) si (ri)aggiornino i valori di `tang_prec` e `tang_ip_prec` in modo tale che siano posti uguali rispettivamente a `tang` e `tang_ip`, in altri termini nelle *variabili locali* `tang_prec` e `tang_ip_prec` vengono memorizzati i valori della tangente trigonometrica e della tangente iperbolica così come sono stati approssimati prima della corrente iterazione del ciclo;

(A42) si incrementi di 1 il valore del contatore m;

(A43) si memorizzi in una variabile locale `coef` il valore del contributo del termine di indice m che compare nella serie di Taylor che definisce la tangente iperbolica;

tale contributo altro non è che `coef` $= \left[2^{2m}\left(2^{2m}-1\right)\mathcal{B}_{2m}x^{2m-1}\right]/\left[(2m)!\right]$, ma nel programma *esso deve essere espresso in modo equivalente grazie a un'opportuna chiamata della function descritta al punto (B) dell'obiettivo 2 e utilizzando opportunamente le variabili* `pot2` , `potx` e `fatt` ;

(A44) si aggiorni il valore di `tang` , aggiungendovi il contributo del termine di indice m che compare nella serie di Taylor, la quale definisce $\tan(x)$ in formula (5); tale contributo altro non è che il prodotto tra i valori delle variabili σ e `coef` ;

(A45) si aggiorni il valore di `tang_ip` , aggiungendovi il contributo del termine di indice m che compare nella serie di Taylor, la quale definisce $\tanh(x)$ in formula (5); tale contributo altro non è che il valore della variabile `coef` ;

(A46) si cambi il segno della variabile σ (si osservi che, dopo l'esecuzione di questa istruzione, si ha che $\sigma = (-1)^m$);

(A47) si aggiorni il valore della variabile `fatt` moltiplicandolo per $(2m+1)(2m+2)$ (si osservi che, dopo l'esecuzione di questa istruzione, si ha che `fatt` $= (2m+2)!$);

(A48) si aggiorni il valore della variabile `pot2` moltiplicandolo per 4 (si osservi che, dopo l'esecuzione di questa istruzione, si ha che `pot2` $= 2^{2(m+1)}$);

(A49) si aggiorni il valore della variabile `potx` moltiplicandolo per x^2 (si osservi che, dopo l'esecuzione di questa istruzione, si ha che `potx` $= x^{2m+1}$);

(A5) *se* il valore di `tang` è diverso da quello di `tang_prec` *oppure* il valore di `tang_ip` è diverso da quello di `tang_ip_prec` , allora si torni a ripetere le operazioni descritte ai precedenti punti (A41)–(A49);

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

(B1) *l'input da tastiera* del valore dell'argomento x ; tale valore inserito in *input* deve essere sottoposto a un test in modo tale che, se $|x|$ *non è minore* di $\pi/10$, allora esso deve essere *reinserito correttamente*;

(B2) *un'opportuna chiamata della function descritta al punto (A)*, in modo da calcolare l'approssimazione numerica di $\tan(x)$ e $\tanh(x)$, dove il valore di x altro non è che quello inserito in *input* così come richiesto al precedente punto (B1);

(B3) la stampa sul video *in formato esponenziale* del valore assoluto della differenza tra il valore di $\tan(x)$ calcolato così come richiesto al precedente punto (B2) e quello che si ottiene evocando la *function* `tan` (che fa parte della libreria matematica del linguaggio **C**);

(B4) la stampa sul video *in formato esponenziale* del valore assoluto della differenza tra il valore di $\tanh(x)$ calcolato così come richiesto al precedente punto (B2) e quello che si ottiene evocando la *function* `tanh` (che fa parte della libreria matematica del linguaggio **C**);

Ovviamente, la verifica numerica del calcolo "simultaneo" (per espansione in serie) della funzione tangente (trigonometrica) e della tangente iperbolica *può considerarsi ben riuscita*, quando entrambi i valori assoluti delle suddette differenze (di cui sono richieste le stampe sullo schermo ai precedenti punti (B3)–(B4)) sono circa dell'ordine di grandezza dell'*errore di macchina.*

Obiettivo (finale) 4:

si integri il programma richiesto dall'obiettivo 3, in modo tale da aggiungere il calcolo approssimato del valore di π , utilizzando la formula (6), dove i numeri di Bernoulli e quelli di Eulero vengono determinati utilizzando, ripettivamente, l'algoritmo di Akiyama–Tanigawa e la definizione (7). A tali scopi si proceda come segue:

(A) si scriva una *function* che ha come solo argomento l'indice intero m; *(alla fine della chiamata)* tale *function* deve restituire il valore del numero di Bernoulli $\mathcal{B}_m$, il quale deve essere calcolato traducendo opportunamente l'algoritmo descritto ai punti (α1)–(α3) in linguaggio di programmazione **C**;

(B) si scriva una *function* con tre argomenti, che sono il massimo indice (di tipo `int`) contenuto nella variabile $\bar{m}$ e due vettori (i cui elementi sono di tipo `double`); la *function* deve essere scritta in modo tale che, $\forall\, m = 0,\, 1,\, \ldots\, ,\, \bar{m}$, l'$m$–esimo elemento del primo vettore (in argomento) sia posto uguale a $\mathcal{B}_m$ grazie a *un'opportuna chiamata della function descritta al punto (A)*, invece, l'm–esimo elemento del secondo vettore (in argomento) sia posto uguale a $\mathcal{E}_m$ grazie a *un'opportuna traduzione della formula* (7) *in linguaggio di programmazione* **C**;

(C) la *main function* deve essere *ampliata* in modo tale che, a sua volta, essa contenga:
   - (C1) *l'input da tastiera* del valore del massimo indice $\bar{m}$; tale valore (di tipo `int`) inserito in *input* deve essere sottoposto a un test in modo tale che, se $\bar{m}$ *non è compreso tra* 2 e 50, allora esso deve essere *reinserito correttamente*;
   - (C2) *la definizione di due vettori* (o, in termini tecnicamente più corretti, *l'allocazione di memoria per due array* 1*–dimensionali*) i cui elementi vanno da quello corrispondente al posto 0 fino a quello di indice $\bar{m}$ e sono tutti di tipo `double`;
   - (C3) *un'opportuna chiamata della function descritta al punto (B)*, in modo tale che, al termine della chiamata stessa, il primo dei due vettori (definiti così come richiesto al precedente punto (C2)) contenga i valori calcolati dei numeri di Bernoulli $\mathcal{B}_0$, $\mathcal{B}_1, \ldots, \mathcal{B}_{\bar{m}}$, invece, il secondo di quei due vettori dovrà contenere i valori calcolati dei numeri di Eulero $\mathcal{E}_0,\, \mathcal{E}_1,\, \ldots,\, \mathcal{E}_{\bar{m}}$;
   - (C4) un ciclo che *per ogni valore pari dell'indice m che sia compreso tra* 2 *e* $\bar{m}$, venga stampata sul video una riga nella quale compaiono ordinatamente i valori delle seguenti quattro quantità:
     m, $\mathcal{B}_m$, $\mathcal{E}_m$ e $\left|\pi - 2\left(2^m - 4^m\right)\mathcal{B}_m/\mathcal{E}_m\right|$.

Alcune osservazioni finali

In linea di principio, l'equazione (6) implica che un chiaro comportamento di convergenza a zero dovrebbe essere evidenziato dai valori che vengono riportati in corrispondenza alla quarta colonna dell'*output* su video, il quale è richiesto così come descritto al punto (C4) dell'obiettivo finale 4. In verità, ciò non avviene perché l'algoritmo di Akiyama–Tanigawa produce un rapidissimo accumulo degli errori (come discusso nella nota[⋆], riportata al termine di pagina 2). L'andamento empirico mostra che la funzione (di variabile discreta) $m \mapsto \left|\pi - 2\left(2^m - 4^m\right)\mathcal{B}_m/\mathcal{E}_m\right|$ ha un minimo in corrispondenza a $m = 12$, laddove i calcoli sono stati effettuati con variabili di tipo `double`, in un programma compilato con `gcc 4.8.2` e mandato in esecuzione su un comune processore di tipo `Intel` con architettura a 64 bit.