Atti Accademia Nazionale dei Lincei

Classe Scienze Fisiche Matematiche Naturali

Rendiconti

Francesco Giacomo Tricomi

Algoritmi iterativi pel calcolo della somma di talune serie

Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti, Serie 8, Vol. **39** (1965), n.6, p. 395–399.

Accademia Nazionale dei Lincei

<http://www.bdim.eu/item?id=RLINA_1965_8_39_6_395_0>

Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti, Accademia Nazionale dei Lincei, 1965.

RENDICONTI

DELLE SEDUTE

DELLA ACCADEMIA NAZIONALE DEI LINCEI

Classe di Scienze fisiche, matematiche e naturali

Seduta dell'11 dicembre 1965

Presiede il Presidente BENIAMINO SEGRE

NOTE DI SOCI

Analisi numerica. — *Algoritmi iterativi pel calcolo della somma di talune serie.* Nota [*] del Socio FRANCESCO GIACOMO TRICOMI.

1. In una precedente Nota [1] ho posto in luce, partendo da un caso concreto, la relazione intercedente fra taluni prodotti infiniti e certi algoritmi iterativi, per mezzo della quale si può procedere al calcolo per via iterativa (che è la più comoda per l'impiego dei moderni calcolatori elettronici) di talune funzioni, per esempio del *seno*. Una relazione analoga, anzi ancora più semplice, si può anche stabilire fra algoritmi iterativi e talune serie, come mostro in questa Nota in cui tale idea viene applicata a tre diversi esempi, ottenendo in due essi dei nuovi metodi, molto efficienti, pel calcolo numerico delle funzioni *gamma* incomplete e delle funzioni *theta* ellittiche.

All'uopo si consideri che, se si ha una serie della forma

$$\mathfrak{A} = \sum_{m=1}^{\infty} f(z_m) \qquad \text{essendo} \quad z_{m+1} = g(z_m)\,, \tag{1}$$

basta porre

$$u_n = \sum_{m=1}^{n} f(z_m) \qquad \text{donde} \quad \mathfrak{A} = \lim_{n\to\infty} u_n \tag{2}$$

(*) Presentata nella seduta dell'11 dicembre 1965.

(1) F. G. TRICOMI, *Un algoritmo iterativo pel calcolo della funzione seno.* In questo stesso volume dei « Rendiconti », pp. 146–150.

per poter procedere al calcolo numerico di $\mathfrak{U}$ per mezzo dell'algoritmo iterativo:

$$\left\{\begin{aligned} z_{n+1} &= g(z_n) \\ u_{n+1} &= u_n + f(z_{n+1}) \end{aligned}\right. \qquad (n = 1, 2, 3, \cdots; u_1 = g(z_1)). \tag{3}$$

che, se le funzioni f e g sono semplici o si ha già pronto un sottoprogramma per il loro calcolo, è di assai facile programmazione ed esecuzione.

Sono lungi del pensare che sia sempre conveniente calcolare per tal via la somma di una data serie, ma vi sono casi in cui il procedimento pare degno di attenzione, come in alcuni dei tre esempi seguenti.

2. Come primo esempio consideriamo il classico sviluppo della funzione *cotangente* in serie di funzioni razionali:

$$\pi \operatorname{cotg}(\pi x) = \frac{1}{x} + \sum_{n=1}^{\infty} \frac{2x}{x^2 - n^2}.$$

In tal caso conviene porre

$$z_n = 1 + \frac{n}{x} \quad , \quad f(z) = \frac{1}{z(2-z)}$$

e si ha

$$\operatorname{cotg}(\pi x) = \frac{1}{\pi x}(1 + 2 \lim_{n\to\infty} u_n),$$

essendo le successive u_n calcolate a partire di $z_1 = 1 + 1/x$ ed $u_1 = f(z_1)$ per mezzo del ben semplice algoritmo iterativo

$$\left\{\begin{aligned} z_{n+1} &= z_n + \frac{1}{x}, \qquad\qquad (n = 1, 2, 3, \cdots) \\ u_{n+1} &= u_n + f(z_{n+1}) = u_n + \frac{1}{z_{n+1}(2 - z_{n+1})}. \end{aligned}\right. \tag{4}$$

Non c'è però da aspettarsi una convergenza molto rapida perché la serie di partenza converge piuttosto lentamente. Lo specchietto seguente condensa una parte dei risultati ottenuti in un esperimento numerico relativo al caso $x = 1/4$, in cui la *cotangente* vale 1. Per brevità si è posto

$$v_n = \frac{1}{\pi x}(1 + 2u_n)$$

sicché, nel caso considerato, il limite cui tende v_n è *uno*.

n	z_n	u_n	v_n
1	5	—0,066 667	1,103 474
10	41	101 352	015 149
20	81	104 253	007 763
30	121	105 252	005 218
40	161	105 758	003 930

La convergenza si potrebbe accelerare con artifizi analoghi a quelli indicati nella Nota precedente, ma non è il caso di indugiarsi su ciò perché questo esempio ha solo valore « teorico ».

3. Il secondo esempio è offerto dalla *funzione gamma incompleta*:

$$\gamma(\alpha, x) = \int_0^x e^{-t} t^{\alpha-1} dt, \qquad (\alpha > 0)$$

che, moltiplicata per e^x, ha lo sviluppo in serie dappertutto convergente

$$e^x \gamma(\alpha, x) = x^\alpha \sum_{n=0}^{\infty} \frac{x^n}{\alpha(\alpha+1)\cdots(\alpha+n)} \,.$$

In questo caso conviene assumere

$$f(z) = z \quad , \quad z_n = \frac{x^n}{\alpha(\alpha+1)\cdots(\alpha+n)}$$

donde segue

$$\frac{z_{n-1}}{z_n} = \frac{\alpha+n}{x} \,.$$

Ciò mostra che conviene modificare leggermente lo schema precedente cercando una relazione (indipendente da n) fra *tre* z_n successive invece che fra *due*; trovando così che

$$\frac{z_n}{z_{n+1}} = \frac{z_{n-1}}{z_n} + \frac{1}{x}$$

cioè

$$z_{n+1} = \frac{x z_n^2}{z_n + x z_{n-1}} \,.$$

Siamo così condotti all'algoritmo iterativo:

$$\text{(5)} \qquad \left\{ \begin{aligned} z_{n+1} &= \frac{x z_n^2}{z_n + x z_{n-1}} \\ u_{n+1} &= u_n + z_{n+1} \end{aligned} \right. \qquad (n = 1, 2, 3, \cdots)$$

che, a partire dai valori iniziali

$$z_0 = \frac{1}{\alpha} \quad , \quad z_1 = u_1 = \frac{x}{\alpha(\alpha+1)} \,,$$

conduce al calcolo della funzione $\gamma(\alpha, x)$ con la formula

$$\text{(6)} \qquad \gamma(\alpha, x) = e^{-x} x^\alpha \lim_{n\to\infty} u_n \,.$$

In tal caso, oltre al fatto che ci si può aspettare una buona convergenza, si ha il vantaggio grandissimo che l'algoritmo (5) è *indipendente da* α, sicché un unico programma serve pel calcolo di $\gamma(\alpha, x)$ qualunque sia $\alpha > 0$.

Ho fatto un esperimento numerico supponendo $\alpha = 1/2$, nel qual caso la funzione $\gamma(\alpha, x)$ si riduce alla *funzione degli errori*. Precisamente si ha

$$\varphi(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = \frac{1}{\sqrt{\pi}} \gamma\left(\frac{1}{2}, x^2\right).$$

Supponendo $x = 1$ e indicando con φ_n il valore approssimato di $\varphi(1)$ fornito dell'*n–esima* iterazione si ha

n	z_n	u_n	φ_n
0	2,0000 0000	2,0000 0000	0,4151 0750
1	1,3333 3333	3,3333 3333	6918 4583
2	0,5333 3333	3,8666 6667	8025 4116
4	0338 6243	4,0529 1005	8411 9667
6	0009 4720	4,0600 1406	8426 7114
8	0000 1486	4,0601 5521	8427 0043
10	0000 0015	4,0601 5692	8427 0079

Come si vede, *dieci* iterazioni forniscono già 8 decimali esatti perché è

$$\varphi(1) = 0,8427\ 0079\ 29\ .$$

4. Come terzo ed ultimo esempio vogliamo considerare la terza delle funzioni *theta* ellittiche che, nel caso ben frequente in cui il rapporto dei periodi è immaginario puro: $\tau = it$ può utilmente rappresentarsi con la serie

$$\vartheta_3(v \mid it) = \frac{1}{\sqrt{t}} \sum_{n=-\infty}^{\infty} e^{-\frac{\pi}{t}(n-v)^2}, \qquad (t > 0).$$

In questo caso conviene porre

$$f(z) = e^{-z^2} \quad , \quad z_m = \sqrt{\frac{\pi}{t}}(v - m) \quad , \quad z'_m = \sqrt{\frac{\pi}{t}}(v + m)$$

$$u_n = \sum_{m=-n}^{n} f(z_m)$$

in modo da ottenere l'algoritmo iterativo:

$$(7) \qquad \begin{cases} z_n = z_{n-1} - \sqrt{\dfrac{\pi}{t}} \quad , \quad z'_n = z'_{n-1} + \sqrt{\dfrac{\pi}{t}} \\ u_n = u_{n-1} + f(z_n) + f(z'_n) \end{cases} \qquad (n = 1, 2, 3, \cdots)$$

che, a partire dai valori iniziali

$$z_0 = z'_0 = \sqrt{\frac{\pi}{t}}\, v \quad , \quad u_0 = f(z_0)\, ,$$

permette di calcolare il valore di ϑ_3 con la formula

$$\vartheta_3(v \mid it) = \frac{1}{\sqrt{t}} \lim_{n\to\infty} u_n .$$

Stavolta è da aspettarsi una convergenza rapidissima, e gli esperimenti numerici che ho condotto in condizioni volutamente sfavorevoli lo confermano. Precisamente ho considerato, ponendo $v = 0$, i due casi in cui il corrispondente modulo k delle funzioni di Jacobi ha i valori sin 85° oppure sin 5°; casi a cui rispettivamente corrispondono:

$(k = \sin 85^\circ)$ $\quad t = 0{,}4107\ 2497\ 34 \quad ; \quad q = 0{,}2751\ 7980\ 49 \; ;$

$$\vartheta_3(0) = 1{,}5618\ 4593\ 34$$

$(k = \sin 5^\circ)$ $\quad t = 2{,}4347\ 1925\ 20 \quad ; \quad q = 0{,}0004\ 7656\ 99 \; ;$

$$\vartheta_3(0) = 1{,}0009\ 5313\ 98 .$$

Nel primo caso basta già *una sola* iterazione per avere ϑ_3 con 10 decimali esatti, come risulta dal seguente compendio dei calcoli eseguiti, come i precedenti, col calcolatore IBM 610:

n	$z_n^2 = z_n'^2$	u_n	$u_n/\sqrt{t}$
0	0,0000 0000 00	1,0000 0000 00	1,5603 5869 34
1	7,6488 9611 56	1,0009 5313 98	1,5618 4593 34

Nel secondo caso si ottengono 8 decimali esatti con *tre* iterazioni:

n	$z_n^2 = z_n'^2$	u_n	$u_n/\sqrt{t}$
0	0,0000 0000 00	1,0000 0000 00	0,6408 7828 28
1	1,2903 3055 91	1,5503 5960 72	0,9935 9180 28
2	5,1613 2223 63	1,5618 2783 28	1,0009 4153 96
3	11,6129 7503 16	1,5618 4592 87	1,0009 5313 68
4	20,6452 8894 51	1,5618 4593 08	1,0009 5313 82

ZUSAMMENFASSUNG. — Man zeigt wie die Summe einer unendlichen Reihe, mittels eines iterativen Algorithmus manchmals berechnet werden kann, was beim Einsatz der modernen Rechnenanlagen sehr vorteilhaft ist.

Von den drei betrachten Beispielen scheinen denen der unvollständigen Gammafunktion und der elliptischen Thetafunktionen praktisch brauchbar zu sein.