

Relazioni di Laboratorio di Fisica Computazionale

Carlo Sana

2 giugno 2012

Indice

1	Integrazione numerica	2
1.1	Metodi deterministici	2
1.1.1	Newton-Cotes	2
1.2	Metodi MonteCarlo	7
2	Integrali di cammino nell'oscillatore armonico	8
3	Runge-Kutta IV	10

1

Integrazione numerica

Si vuole calcolare il valore dell'integrale definito di una funzione reale di variabile reale in una dimensione:

$$I = \int_{x_{min}}^{x_{max}} f(x) dx$$

Sono stati implementati diversi metodi di integrazione numerica, deterministici e MonteCarlo, che verranno poi confrontati fra di loro.

1.1 Metodi deterministici

Le prime routine di integrazione che sono state scritte sono un'implementazione delle formule di Newton-Cotes (al primo e secondo ordine) e del metodo delle quadrature gaussiane.

Per aumentare la precisione del calcolo, il dominio di integrazione viene suddiviso in sottointervalli. La larghezza di ogni intervallo è uniforme ed è possibile scegliere il numero di intervalli in cui si vuole dividere il dominio di integrazione prima di chiamare le funzioni. Per ottenere la stima dell'integrale è necessario sommare le stime degli integrali ottenute per i sottointervalli.

Nel nostro caso n sarà il numero di sottointervalli. Definiamo così in modo naturale una partizione dell'insieme di integrazione:

$$h = \frac{x_{max} - x_{min}}{n} \quad \implies \quad a_i = x_{min} + ih \quad \text{per } i = 0, 1, \dots, n$$

Nel caso gli estremi di integrazione sono scambiati, ossia $x_{min} > x_{max}$, l'integrazione avviene comunque correttamente, visto che in questo caso h sarà negativo.

1.1.1 Newton-Cotes

Le formule di Newton-Cotes si ottengono interpolando la funzione integranda con polinomi di Lagrange. Il polinomio di Lagrange j -esimo di grado n è definito come:

$$l_j^n(x) = \prod_{i=0, i \neq j}^n \frac{x - x_i}{x_j - x_i}$$

Come si può vedere è un polinomio di grado n , definito in base alla partizione scelta per l'intervallo, con la proprietà:

$$l_j^n(x_i) = \delta_{ij}$$

E' ora immediato costruire un polinomio $P(x)$ tale che $P(x_i) = f(x_i) \quad \forall 0 < i < n$.

Questo polinomio è il seguente:

$$P(x) = \sum_{i=0}^n f(x_i) l_i^n(x)$$

La stima dell'integrale diventa così:

$$I = \int_{x_{min}}^{x_{max}} P(x) dx = \int_{x_{min}}^{x_{max}} \sum_{i=0}^n f(x_i) l_i^n(x) dx = \sum_{i=0}^n f(x_i) \omega_i$$

Si dimostra inoltre che, con un cambio di variabile:

$$\omega_j = \int_{x_{min}}^{x_{max}} l_j^n(x) dx = \int_0^n \prod_{i \neq j} \frac{z - i}{j - i} dz$$

indipendente dall'intervallo di integrazione.

Questa è ovviamente una stima dell'integrale e si dimostra che l'errore, utilizzando $n+1$ punti è uguale a :

$$E_n = \frac{1}{(n+1)!} \int_{x_{min}}^{x_{max}} f^{n+1}(\xi) \prod_{i=0}^n (x - x_i) dx$$

dove ξ è un punto interno all'intervallo. L'errore è facilmente sovrastimabile, valutando il massimo della derivata $n+1$ -esima all'interno dell'intervallo. L'errore sulla stima dell'integrale, però, non viene calcolato dalla routine d'integrazione. Ciò deriva dal fatto che esso è stimabile analiticamente, essendo necessario solo calcolare la derivata dell'integranda e valutarne il massimo nell'intervallo.

Newton-Cotes:1°ordine

Questo metodo consiste nell'approssimare la funzione fra due punti a_i e a_{i+1} con un segmento. L'area si ottiene calcolando l'area del trapezio sotteso da questo segmento, oppure applicando le formule di Newton-Cotes, ponendo $n = 1$:

$$\omega_0 = \frac{1}{2} \quad \omega_1 = \frac{1}{2}$$

In questo caso la stima analitica dell'errore diventa:

$$|E_1(f)| = \frac{h^3}{12} f''(\xi) \leq \frac{h^3}{12} \max [f''(x), x \in [x_{min}, x_{max}]]$$

Newton-Cotes:2°ordine

In questo caso, l'approssimazione viene fatta con polinomi di grado 2, ossia parabole. I pesi ω_i valgono:

$$\omega_0 = \frac{1}{6} \quad \omega_1 = \frac{2}{3} \quad \omega_2 = \frac{1}{6}$$

La stima dell'errore si può scrivere come:

$$|E_2(f)| = \frac{h^5}{90} f^{(4)}(\xi) \leq \frac{h^5}{90} \max [f^{(4)}(x), x \in [x_{min}, x_{max}]]$$

Quadrature gaussiane

Nel caso delle quadrature gaussiane, i punti della partizione non vengono più scelti equidistanti, ma vengono scelti in maniera più opportuna: sono gli zeri del polinomio ortogonale scelto.

E' fondamentale l'uso di polinomi ortogonali in un certo intervallo $[a, b]$ con il peso $\omega(x)$:

$$\int_a^b \omega(x) P_n(x) P_m(x) dx = \delta_{m,n}$$

Si può dimostrare che la stima dell'integrale è:

$$I = \int_a^b f(x) dx = \sum_{i=0}^n \omega_i f(x_i)$$

dove ω_i , nel caso dei polinomi di Legendre, sono gli stessi pesi definiti per i polinomi di Lagrange costruiti sull'insieme degli zeri del polinomio di Legendre considerato. Considerando che un polinomio ortogonale di grado n in $[a, b]$ ha n zeri in $[a, b]$, anche i polinomi di lagrange saranno di grado n . Inoltre x_i sono gli zeri del polinomio ortogonale considerato. Gli zeri del polinomio sono stati inseriti manualmente, lasciando però al calcolatore il compito di calcolarne il valore in virgole mobile, per aumentarne la precisione. All'interno dell'algoritmo, è necessario effettuare un cambio di variabile in modo da mappare l'intervallo di integrazione nell'intervallo in cui il polinomio è ortogonale. Nel nostro caso tale intervallo è $[-1, 1]$, dato che è stato usato un polinomio di Legendre.

Tale cambio di variabile è:

$$x' = 2 \frac{x - x_{min}}{x_{max} - x_{min}} - 1 \quad dx' = 2 \frac{dx}{x_{max} - x_{min}}$$

Utilizzo

Il programma "integral" si occupa di utilizzare le librerie appena discusse per stimare il valore dell'integrale di una funzione, confrontando i tre metodi discussi sopra. Per testare le routine ho utilizzato come funzioni da integrare:

$$f_1(x) = \log(1+x) \Rightarrow \text{primitiva} \quad -x + \log(1+x) + x \log(1+x)$$

$$f_2(x) = x^9 - x^7 + 3 \Rightarrow \text{primitiva} \quad \frac{x^{10}}{10} - \frac{x^8}{8} + 3x$$

In questo modo si è potuto confrontare i tre metodi di integrazione con il valore vero dell'integrale, valutandone lo scostamento dal valore vero e confrontandolo con la stima dell'errore calcolata analiticamente. È riportata nella tabella seguente una serie di risultati ottenuti variando il numero di intervalli, ma mantenendo costanti gli estremi di integrazione. In questo caso l'intervallo di integrazione è stato $[1, 2]$. La prima tabella si riferisce alla funzione integranda di tipo logaritmico, la seconda a quella di tipo polinomiale. La stima analitica dell'errore è stata calcolata valutando il massimo della derivata in tutto l'intervallo di integrazione. Questa risulta essere una sovrastima della stima dell'errore. Tali valori sono stati calcolati con *Wolfram Mathematica* e risultano essere:

funzione	Max f'	Max f''
f_1 (Log)	0.5	-0.0740742
f_2 (Poly)	1856	90048

Log

Numero intervalli	Newton-Cotes I	Stima analitica errore
10	1.388645e-04	4.166667e-04
20	3.472070e-05	1.041667e-04
40	8.680460e-06	2.604167e-05
80	2.170133e-06	6.510417e-06
160	5.425343e-07	1.627604e-06
320	1.356337e-07	4.069010e-07
640	3.390842e-08	1.017253e-07
1280	8.477104e-09	2.543132e-08
2560	2.119278e-09	6.357829e-09
5120	5.298200e-10	1.589457e-09
10240	1.324560e-10	3.973643e-10
20480	3.310918e-11	9.934107e-11
40960	8.281043e-12	2.483527e-11
81920	2.071121e-12	6.208817e-12

Da questo confronto si nota come l'errore scala come previsto analiticamente e rimane sempre inferiore alla stima fatta analiticamente.

Log		
Numero intervalli	Newton-Cotes II	Stima analitica errore
10	-6.101824e-09	-8.230467e-08
20	-3.816787e-10	-5.144042e-09
40	-2.386025e-11	-3.215026e-10
80	-1.491696e-12	-2.009391e-11
160	-9.336976e-14	-1.255870e-12
320	-6.217249e-15	-7.849185e-14
640	-2.220446e-16	-4.905740e-15
1280	-1.443290e-15	-3.066088e-16
2560	-6.661338e-16	-1.916305e-17
5120	-2.664535e-15	-1.197691e-18
10240	1.443290e-15	-7.485566e-20
20480	2.220446e-15	-4.678479e-21
40960	2.220446e-15	-2.924049e-22
81920	-3.108624e-15	-1.827531e-23

in questo caso invece l'errore scala correttamente fino a quando raggiunge il valore di $10^{-15}/10^{-16}$. A quel punto smette di diminuire all'aumentare degli intervalli considerati ed inizia ad oscillare, cambiando anche di segno.

Log	
Numero intervalli	quadratura gaussiana
10	2.220446e-16
20	2.220446e-16
40	2.220446e-16

80	2.220446e-16
160	5.551115e-16
320	1.221245e-15
640	-3.330669e-16
1280	1.332268e-15
2560	-6.661338e-16
5120	-2.775558e-15
10240	-1.443290e-15
20480	-2.220446e-15
40960	-2.109424e-15
81920	2.997602e-15

Poly		
Numero intervalli	Newton-Cotes I	Stima analitica errore
10	1.541035e+00	1.546667e+00
20	3.860018e-01	3.866667e-01
40	9.654698e-02	9.666667e-02
80	2.413966e-02	2.416667e-02
160	6.035096e-03	6.041667e-03
320	1.508785e-03	1.510417e-03
640	3.771970e-04	3.776042e-04
1280	9.429930e-05	9.440104e-05
2560	2.357483e-05	2.360026e-05
5120	5.893707e-06	5.900065e-06
10240	1.473427e-06	1.475016e-06
20480	3.683565e-07	3.687541e-07
40960	9.208941e-08	9.218852e-08
81920	2.302234e-08	2.304713e-08

in questo caso l'andamento è simile al caso della funzione logaritmica interpolata con il metodo dei trapezi. Si può notare che in questo caso la differenza fra l'errore stimato e l'errore di integrazione è molto piccola. In ogni caso l'errore stimato è sempre maggiore dell'errore di integrazione.

Poly		
Numero intervalli	Newton-Cotes II	Stima analitica errore
10	9.908609e-04	1.000533e-01
20	6.203492e-05	6.253333e-03
40	3.878841e-06	3.908333e-04
80	2.424535e-07	2.442708e-05
160	1.515373e-08	1.526693e-06
320	9.471250e-10	9.541829e-08
640	5.921663e-11	5.963643e-09
1280	3.652190e-12	3.727277e-10

2560	1.847411e-13	2.329548e-11
5120	4.263256e-14	1.455968e-12
10240	-1.705303e-13	9.099798e-14
20480	4.263256e-14	5.687374e-15
40960	-3.552714e-13	3.554608e-16
81920	4.263256e-14	2.221630e-17

anche in questo caso l'andamento è simile alla funzione logaritmica, anche se l'errore inizia ad oscillare ad un valore di $10^{-13}/10^{-14}$.

Poly	
Numero intervalli	quadratura gaussiana
10	0.000000e+00
20	0.000000e+00
40	1.421085e-14
80	1.421085e-14
160	2.842171e-14
320	0.000000e+00
640	2.842171e-14
1280	-2.842171e-14
2560	2.842171e-14
5120	-2.842171e-14
10240	1.705303e-13
20480	-4.263256e-14
40960	3.552714e-13
81920	-2.842171e-14

Dall'analisi di questi dati è possibile trarre alcune conclusioni riguardo i diversi metodi di integrazione deterministica utilizzati.

È banale notare come la precisione dell'integrazione con le formule di Newton-Cotes al II° ordine sia molto maggiore di quelle al I° ordine. Inoltre, si nota come l'errore scali nella maniera prevista fino a quando diventa troppo "piccolo" e inizia ad oscillare attorno a zero. Questo può essere dovuto alla precisione con cui sono salvati i numeri nel calcolatore, nonostante siano state usate variabili di tipo *double* in tutto il programma. Ciò si verifica quando è necessario sommare numeri molto piccoli, in questo caso il valore dell'integrale in ogni bin, a numeri molto più grandi, ossia l'integrale vero e proprio.

1.2 Metodi MonteCarlo

2

Integrali di cammino nell'oscillatore armonico

Si è risolto l'oscillatore armonico quantistico monodimensionale attraverso l'utilizzo degli integrali di cammino sul reticolo. È stato scelto un approccio non-deterministico attraverso l'integrazione MonteCarlo. Il tempo è stato discretizzato in N istanti e, ponendo T come istante finale e 0 come istante iniziale, abbiamo il passo reticolare temporale $a = \frac{T}{N}$.

Integrali di cammino

Nel formalismo del *path integral* sul reticolo è fondamentale introdurre il concetto di azione. Nel nostro caso sarà importante la definizione di azione euclidea:

$$S_E = a \sum_{i=0}^{n-1} \mathcal{L}(x_i, x_{i+1})$$

dove

$$\mathcal{L}(x_i, x_{i+1}) = \frac{m}{2} \left(\frac{x_{i+1} - x_i}{a} \right)^2 + V(x)$$

Si ricava un equivalente alla funzione di partizione classica definita come:

$$Z_a(0, T) = \left(\frac{m}{2\pi a} \right)^{N/2} \int \prod_{i=1}^{n-1} dx_i e^{-S_E}$$

e il correlatore fra due operatori di posizione è uguale a:

$$\langle x_l x_k \rangle = \frac{\int \prod_{i=1}^{n-1} dx_i x_l x_k e^{-S_E}}{Z_a}$$

Si dimostra che:

$$\langle x_l x_k \rangle = 2 |\langle \tilde{E}_0 | \hat{x} | \tilde{E}_1 \rangle|^2 \exp \left(-\frac{N a}{2} (\tilde{E}_1 - \tilde{E}_0) x \right) \cosh \left[a \left(\frac{N}{2} - |l - k| \right) (\tilde{E}_0 - \tilde{E}_1) \right]$$
$$\langle E_0 | \hat{x} | E_1 \rangle = \frac{1}{\sqrt{2m\tilde{\omega}}}$$

Nel nostro caso, si è posto $N = 32$, ma l'algoritmo è indipendente da N .

Ciò che viene calcolato dall'algoritmo sono i valori medi $\langle x_l x_k \rangle$ per ogni valore di $|l - k|$ e i relativi errori. Da essi siamo in grado di estrarre il valore di $\langle E_0 | \hat{x} | E_1 \rangle$ e $\tilde{E}_1 - \tilde{E}_0$. Gli errori sulle grandezze secondarie sono stati calcolati con il metodo *cluster jackknife*.

Algoritmo Metropolis

Per calcolare $\langle x_l x_k \rangle$ con un metodo MonteCarlo è necessario riuscire ad estrarre numeri casuali secondo la *pdf* e^{-S_E} . L'algoritmo *Metropolis-Hastings* permette di estrarre numeri casuali secondo una *pdf* qualsiasi a partire da un generatore “piatto”.

Più in generale esso permette, dato uno spazio di configurazioni S e una *pdf* $P(s) : S \rightarrow \mathbb{R}$, di estrarre configurazioni del sistema compatibili con la *pdf* voluta. Ciò è particolarmente utile in quanto non è sempre possibile trovare un cambio di coordinate che permette di ottenere la distribuzione voluta a partire da una distribuzione piatta, come è stato fatto nell'integrazione MonteCarlo discussa in 1.2.

L'algoritmo si basa sul metodo del rigetto. Poniamo di avere uno stato s nello spazio delle configurazioni. A questo punto:

- si estrae una nuova configurazione del sistema s' con il generatore di numeri random “piatto”.
- si calcola il rapporto $\frac{P(s')}{P(s)}$.
- si accetta il nuovo stato estratto s' con probabilità pari a $\min \left[1, \frac{P(s')}{P(s)} \right]$.

Si dimostra che gli stati del sistema vengono estratti con la *pdf* da noi cercata, a patto di attendere che il sistema si termalizzi. Questo è dovuto al fatto che la *pdf* di estrazione delle configurazioni approssima la distribuzione P solo in regime asintotico. È possibile stimare il tempo di termalizzazione dell'algoritmo graficando l'andamento dell'azione euclidea S_E in funzione del tempo markoviano.

3

Runge-Kutta IV

L'implementazione di questo metodo di risoluzione delle equazioni differenziali ordinarie è valida per sistemi di equazioni differenziali di ordine 2: ossia riconducibili a un sistema di 2 equazioni differenziali.

Nel nostro caso è stata risolta un'equazione newtoniana, ossia della forma

$$\ddot{x} = f(x, t) \iff \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x, t) \end{cases}$$

Il sistema risolto in questo caso è un pendolo smorzato con forzante esterna sinusoidale, ossia risolvibile la seguente equazione differenziale:

$$\ddot{\theta} = f(\theta, \dot{\theta}, t) = -\frac{g}{R} \theta - b \dot{\theta} + Q \cos(\omega t)$$

L'algoritmo è implementato, sostanzialmente in questa funzione. Essendo un'implementazione in dimensione due, riceve come argomento 2 vettori in cui saranno salvati gli incrementi, le due variabili spaziali, e le due funzioni date dal sistema di equazioni differenziali. Nel nostro caso avremo:

$$\begin{cases} x_1 = \theta \\ x_2 = \dot{\theta} \end{cases} \implies f_1 = x_2 \quad f_2 = f(x_1, x_2, t)$$

Nella funzione è utilizzata la variabile di preprocessore H che rappresenta il passo dell'incremento infinitesimo nel tempo.

Un'ultima funzione, infine, valuta l'incremento effettivo della variabile, a seconda dei valori calcolati dalla funzione riportata sopra e immagazzinati in uno dei due vettori $k1$ o $k2$.

I pesi sono definiti dal metodo Runge-Kutta IV.