

MAURIZIO DAPOR (*)

IL METODO DI MONTE CARLO

ABSTRACT - DAPOR M., 2006 - The Monte Carlo Method.

Atti Acc. Rov. Agiati, a. 256, 2006, ser. VIII, vol. VI, B: 5-14.

In modern physics we are interested in systems with many degrees of freedom. Let us consider, for example, the number of atoms in a solid, the number of electrons in an atom, or the number of electrons of a beam interacting with the many atoms and electrons of a solid. In many situations, these systems can be described by the calculations of integrals of very high dimension. The Monte Carlo method gives us a very accurate way to calculate definite integrals of high dimension: it evaluates the integrand at a random sampling of abscissa. The Monte Carlo method is also used for evaluating the many physical quantities necessary to the study of the interactions of particle-beams with solid targets.

KEY WORDS - Monte Carlo Method.

RIASSUNTO - DAPOR M., 2006 - Il metodo di Monte Carlo.

Nella fisica moderna siamo interessati a sistemi con molti gradi di libertà. Consideriamo, ad esempio, il numero di atomi in un solido, il numero di elettroni in un atomo, o il numero di elettroni di un fascio che interagisce con i moltissimi atomi ed elettroni di un solido. In molte situazioni, questi sistemi possono essere descritti mediante il calcolo di integrali multipli in spazi con un numero molto elevato di dimensioni. Il metodo di Monte Carlo fornisce un modo assai accurato per calcolare questo genere di integrali: si valuta l'integrando per numerosi valori di un campione casuale delle ascisse. Il metodo di Monte Carlo viene usato anche per valutare le grandezze fisiche necessarie allo studio delle interazioni di fasci di particelle con bersagli solidi.

PAROLE CHIAVE - Metodo di Monte Carlo.

(*) Istituto Trentino di Cultura, Centro per la ricerca scientifica e tecnologica, Povo di Trento.

INTRODUZIONE

Il caso ha svolto e svolge un ruolo fondamentale sia consentendo la realizzazione di importanti processi quali l'evoluzione della vita ed il poderoso sviluppo dell'intelligenza [1,2] sia in rilevanti ambiti matematici, scientifici ed applicativi [3,4]. Nelle scienze della vita, la singolare cooperazione tra le mutazioni casuali e le regole della selezione naturale, qualora operi su intervalli di tempo molto grandi, può generare - e ha generato, almeno in un caso ben noto a tutti - informazione biologica: informazione cioè provvista di significato nell'ambiente in cui si colloca. È ben chiaro che i dati genetici sono simboli a cui è annessa una semantica: essi sono *compresi* dagli amminoacidi.

Le leggi del caso permettono anche di svolgere calcoli e simulazioni che trovano numerose applicazioni sia in ambito strettamente matematico, sia nel calcolo scientifico [5,6]: e i settori di interesse (probabilità, statistica, fisica della materia, fisica biomedica, studio dell'andamento dei mercati finanziari, modelli economici e di evoluzione sociale, sistemi complessi, problemi meteorologici, solo per menzionarne alcuni) sono talmente numerosi e di tali portata ed importanza da rendere ovvia un'esigenza che potrebbe forse, ad una prima e superficiale osservazione, apparire un po' strana: quella cioè di saper generare con un computer sequenze di numeri casuali.

La parola caso è - da sempre - accompagnata da un'ombra di sospetto, caratterizzata da una valenza prevalentemente negativa. Si associa la casualità al disordine, all'assenza di regole e ci si interroga, pertanto, sul perché di questa singolare propensione di fisici e matematici a voler simulare sequenze di numeri che approssimino quelle generate da eventi autenticamente aleatori. Inoltre il computer è una macchina deterministica e i processi genuinamente casuali sono legati a fenomeni assai differenti da quelli riproducibili mediante algoritmi e codici di calcolo. In verità nessun programma per computer è in grado di generare una sequenza di numeri autenticamente casuali. Si parla piuttosto di sequenze di numeri *pseudo-casuali* proprio per identificare quelle serie di numeri che opportuni programmi per computer generano al fine di simulare successioni di eventi genuinamente casuali.

UTILIZZO DEI NUMERI CASUALI

Supponiamo di dover calcolare l'area di una superficie chiusa, di forma complicata. Se possediamo un generatore di numeri casuali di-

istribuiti uniformemente, non dobbiamo fare altro che circondare la nostra curva con un quadrato di lato noto. La superficie di area ignota sia cioè completamente contenuta nel quadrato. Generiamo ora un grande numero di punti casuali che siano uniformemente distribuiti all'interno del quadrato. Ogni volta che un punto casuale cade all'interno della superficie di area ignota aggiorniamo un contatore, sommandovi 1. Quando il numero di punti prodotti è molto grande, il rapporto tra quelli caduti all'interno della superficie ed il numero totale di punti generati sarà prossimo al rapporto tra l'area della superficie e quella (nota) del quadrato.

Il programma di calcolo necessario per effettuare questa semplice applicazione è piuttosto elementare. Eccolo espresso nel linguaggio di programmazione C++ ed applicato al caso del calcolo di un'area ben nota, quella di un circolo di raggio unitario:

```
#include<iostream.h>
#include<conio.h>

double rndm()
{
    return rand()/(RAND_MAX+1.0);
}

main()
{
    int n,l,i;
    double x,y;

    cout << "Numero di punti = ";
    cin >> n;

    i=0;
    for(l=1;l<=n;++l)
    {
        x=rndm();
        y=rndm();
        if((x*x+y*y)<1.0) i=i+1;
    }
    cout << "Area = " << 4.0*i/n;
    getch();
}
```

Se si prova a utilizzare questo programma imponendo valori crescenti al numero n di punti casuali, si ottengono misure A della superficie sempre più prossime a quella corretta che è, come il lettore sa bene, pari a $\pi=3.14159\dots$: con $n=1000$ ricaviamo $A=3.112$, con $n=10.000$ il nostro calcolo fornisce $A=3.114$, con $n=100.000$ si ottiene $A=3.138$, con $n=1000.000$ il programma fornisce il valore $A=3.139$ e così via. Ovviamente, superato un dato valore di n – che dipende dal particolare generatore di numeri casuali utilizzato – non si osservano ulteriori miglioramenti nel calcolo: la ragione è legata al fatto che il generatore di numeri casuali si limita a simulare ma non genera numeri autenticamente casuali e uniformemente distribuiti nell'intervallo $(0,1)$. Pertanto, allorché il numero n diventa molto grande, entrano in gioco questioni di periodicità e altre anomalie che si ripercuotono sul calcolo riducendo sempre più i margini di miglioramento. Chiarito questo punto, è evidente che il metodo descritto può essere utilizzato come una tecnica numerica per il calcolo delle aree – a priori incognite – di superfici di forma qualunque. La cosa non sarebbe tanto interessante, giacché esistono procedure di integrazione migliori di quella appena descritta, se non fosse che questo metodo si applica altrettanto bene al caso di «iper-superfici» localizzate in spazi caratterizzati da un numero arbitrariamente elevato di dimensioni. Non è rilevante il fatto che il nostro cervello non ci permetta di visualizzare spazi con più di tre dimensioni: in fisica e in matematica accade spesso di dover calcolare quantità definite su spazi con un elevato numero di dimensioni.

Sistemi con un grande numero di gradi di libertà si incontrano molto spesso, ad esempio, nella meccanica statistica. Per descrivere questi sistemi si fa ricorso alla valutazione di integrali multipli con un grande numero di dimensioni. Un caso importante è rappresentato dal calcolo della funzione di partizione di un gas contenente N atomi alla temperatura T . È stato mostrato che, per un valore di N assai modesto (20), un computer capace di effettuare 10 milioni di calcoli al secondo impiegherebbe, per calcolare la funzione di partizione con i metodi di integrazione numerica tradizionali, un tempo di oltre mille volte superiore all'età stimata dell'universo [7]. Con il metodo appena descritto, invece, poiché il tempo di calcolo non dipende dal numero di dimensioni, esso rimane ragionevolmente contenuto anche per casi come questo.

Questa procedura di calcolo è nota come *metodo di Monte Carlo*. Il metodo di Monte Carlo (che deve il suo nome alle case da gioco della ben nota località) è uno strumento matematico, usato per calcolare il valore medio di molte grandezze, che utilizza generatori di numeri pseu-

do-casuali: esso è, in sostanza, un metodo di integrazione numerica per la valutazione di integrali multipli. Qualora il numero di dimensioni superi il quattro, il metodo di Monte Carlo è la procedura numerica migliore per il calcolo degli integrali multipli [7].

GENERAZIONE DI NUMERI PSEUDO-CASUALI

Nell'esempio illustrato poco sopra per il calcolo di π abbiamo utilizzato un generatore di numeri pseudo-casuali che si trova già codificato nel linguaggio di programmazione C++: si tratta del comando `rand()` che si trova nella procedura `randm()`. Si noti che, al fine di ottenere numeri pseudo-casuali compresi tra 0 e 1, nella procedura `randm()` abbiamo diviso `rand()` per il numero casuale massimo generato dal simulatore.

Si pone la questione di come si possano scrivere algoritmi di generazione di numeri pseudo-casuali. Non forniremo qui un'analisi approfondita: ci limiteremo a suggerire alcune idee di carattere molto generale relative ai metodi solitamente utilizzati.

L'algoritmo più comunemente utilizzato per la generazione di sequenze di numeri pseudo-casuali mediante computer consiste nel far crescere l'intera successione da un «seme»: in altre parole, s'inizia con un numero intero positivo qualunque (il seme) e si calcolano poi i successivi numeri casuali mediante un'equazione che fa dipendere ogni nuovo numero da quello corrente. Così ogni nuovo numero della sequenza di numeri casuali è calcolabile conoscendo il precedente.

Supponiamo che r_n sia l' n -esimo numero casuale generato. Allora il numero pseudo-casuale successivo, che indichiamo con r_{n+1} , si ottiene da

$$r_{n+1} = (a r_n + b) \bmod m, \quad (1)$$

dove a , b ed m sono tre numeri interi. Ricordo che con l'operazione «mod» si indica il calcolo del resto della divisione tra numeri interi. Questo significa che r_{n+1} è uguale al resto della divisione del numero intero $(a r_n + b)$ per il numero intero m . Uno dei problemi delle sequenze di numeri pseudo-casuali è ovviamente costituito, come già constatato poco sopra, dalla periodicità che, fatalmente, le affligge: in effetti, ogni sequenza inevitabilmente ripeterà se stessa con un periodo che non sarà mai più grande del modulo m . Scegliendo opportunamente i valori dei tre numeri «magici» a , b ed m si possono ottenere sequenze corrispondenti al massimo periodo, che è ovviamente uguale a m . Se si costruisce una suc-

cessione di numeri casuali di questo tipo, tutti i numeri interi compresi tra 0 e $m-1$ compariranno in qualche punto della sequenza, qualunque sia il seme iniziale r_0 .

Nella letteratura scientifica sono stati proposti diversi suggerimenti per i valori dei numeri a , b ed m . Sono anche stati effettuati, ovviamente, test statistici per stabilire quali siano i valori dei tre numeri «magici» grazie ai quali meglio si approssima una sequenza di numeri casuali uniformemente distribuiti. Non è questa la sede per una dettagliata analisi. Chi fosse interessato ad approfondire l'argomento, troverà nel capitolo 7 della referenza [8] un ottimo testo di riferimento. Vi si elencano e discutono le più rilevanti proposte che si trovano nella letteratura scientifica relative agli algoritmi di generazione di numeri pseudo-casuali. In particolare per la generazione di una sequenza di numeri pseudo-casuali uniformemente distribuiti che sia simultaneamente semplice, accurata e soddisfacente per la maggior parte delle applicazioni, si suggerisce l'uso del codice definito da $a=16807$, $b=0$ e $m=2147483647$, proposto nel 1969 da Lewis, Goodman e Miller e noto come «Minimal standard». Si noti che la sequenza produce numeri interi compresi tra 0 e $m-1$: per ottenere una sequenza di numeri casuali uniformemente distribuiti tra 0 e 1, è sufficiente dividere per m ognuno dei numeri interi ottenuti (avendo cura, tuttavia, di conservare ed utilizzare sempre il numero intero originale per il calcolo del numero intero successivo).

Naturalmente non è necessario produrre in proprio un generatore di numeri pseudo-casuali, poiché tutti i linguaggi di programmazione già forniscono un comando, come abbiamo visto per il caso del linguaggio C++, per la generazione di numeri casuali.

In ogni modo, una volta che si disponga di un generatore di numeri pseudo-casuali uniformemente distribuiti tra 0 e 1, si pongono questioni assai interessanti come quella di generare variabili casuali che abbiano una distribuzione data, scelta dall'utilizzatore.

GENERAZIONE DI NUMERI CASUALI CON UNA DATA DISTRIBUZIONE

Si può dimostrare che, una volta che si disponga di un generatore di numeri pseudo-casuali uniformemente distribuiti tra 0 e 1, è sempre possibile utilizzarlo per realizzare sequenze di numeri pseudo-casuali corrispondenti a distribuzioni – non più necessariamente uniformi – specificate dall'utente (si veda, per una dettagliata analisi della questione, l'eccellente trattazione di Sobol [5]). Se, ad esempio, indichiamo con la lettera greca μ i numeri casuali distribuiti uniformemente tra 0 e

1 e con la lettera greca η i numeri casuali corrispondenti alla distribuzione di Poisson con valor medio λ definita da

$$p(x) = \exp(-x/\lambda)/\lambda, \quad (2)$$

dove abbiamo indicato con «exp» la funzione esponenziale, allora si può dimostrare che l'equazione

$$\eta = -\lambda \ln \mu, \quad (3)$$

dove abbiamo indicato con «ln» il logaritmo naturale, permette di calcolare una sequenza di numeri casuali distribuiti secondo la legge di Poisson.

Un po' più complicato è il metodo utilizzato per il calcolo di una distribuzione gaussiana, che qui non descriveremo per limitarci piuttosto a scrivere, di seguito, il codice C++ che, a partire dalla funzione `rndm()` – già utilizzata nell'esempio relativo al calcolo di π – permette di calcolare una sequenza di numeri casuali distribuiti secondo una densità di probabilità gaussiana centrata nell'origine e con varianza unitaria:

```
double gauss()
{
    float f,r2,x,y;
    do
    {
        x=2.0*rndm()-1.0;
        y=2.0*rndm()-1.0;
        r2=x*x+y*y;
    }
    while (r2 >= 1.0 || r2 == 0);
    f=sqrt(-2.0*log(r2)/r2);
    return y*f;
}
```

Per approfondire gli aspetti relativi alle distribuzioni di variabili casuali si rimanda alla bibliografia [9]. Per una descrizione dei metodi di calcolo relativi a varie distribuzioni di probabilità, si vedano anche i testi [5,6,8].

USO DEL METODO DI MONTE CARLO NELLA SIMULAZIONE DI PROCESSI FISICI

Sovente il metodo di Monte Carlo viene utilizzato come un algoritmo per la simulazione di processi fisici che coinvolgono grandissimi numeri di particelle perché non esiste, letteralmente, alcuna altra procedura di calcolo in grado di risolvere senza drastiche approssimazioni quel genere di problemi.

In altri termini, esistono ambiti della meccanica statistica che presentano problematiche la cui soluzione numerica è possibile, in pratica, esclusivamente mediante l'utilizzo di sequenze di numeri casuali. Quando il numero di dimensioni è veramente grande, non esiste nessun altro metodo di integrazione numerica che permetta di risolvere i problemi in tempi ragionevoli (nemmeno qualora, in futuro, le potenzialità di calcolo dei computer diventassero superiori a quelle attuali anche di molti ordini di grandezza). Complicati problemi di fisica coinvolgenti grandissimi numeri di particelle possono essere (e, in effetti, sono) affrontati con il metodo di Monte Carlo: si sono realizzate vere e proprie simulazioni numeriche di processi fisici come l'interazione di un fascio di elettroni, o positroni, o neutroni con un solido.

Una delle più interessanti applicazioni del metodo di Monte Carlo consiste nella simulazione dello spettro di perdita di energia degli elettroni retro-diffusi qualora si bombardi un bersaglio solido con un fascio di elettroni di energia cinetica iniziale nota: conoscendo le sezioni d'urto elastica ed anelastica degli elettroni nell'interazione con gli atomi del bersaglio, è possibile calcolare le probabilità di diffusione angolare e di perdita di energia cinetica per ogni collisione tra gli elettroni del fascio incidente e gli atomi del bersaglio. In tal modo si può – letteralmente – seguire la storia di ogni singolo elettrone, percorrendo la sua traiettoria e calcolando le sue perdite di energia, l'eventuale punto di uscita dalla superficie del bersaglio e la sua energia finale. Mediando su un grande numero di traiettorie, si può ottenere uno spettro come quello rappresentato in Figura 1 relativo alla distribuzione energetica di elettroni retro-diffusi da un bersaglio di diossido di silicio. La Figura presenta il cosiddetto picco elastico, una gaussiana centrata attorno all'energia iniziale E_0 del fascio (in questo caso 2000 eV) corrispondente alla frazione di elettroni che escono dalla superficie senza aver subito collisioni anelastiche ⁽¹⁾, ed un certo numero di altri picchi corrispondenti alle perdite di energia dovute alle interazioni anelastiche tra elet-

⁽¹⁾ In Figura 1, in realtà, il picco elastico centrato a 2000 eV non è stato rappresentato in tutta la sua estensione per evitare di appiattire il resto dello spettro.

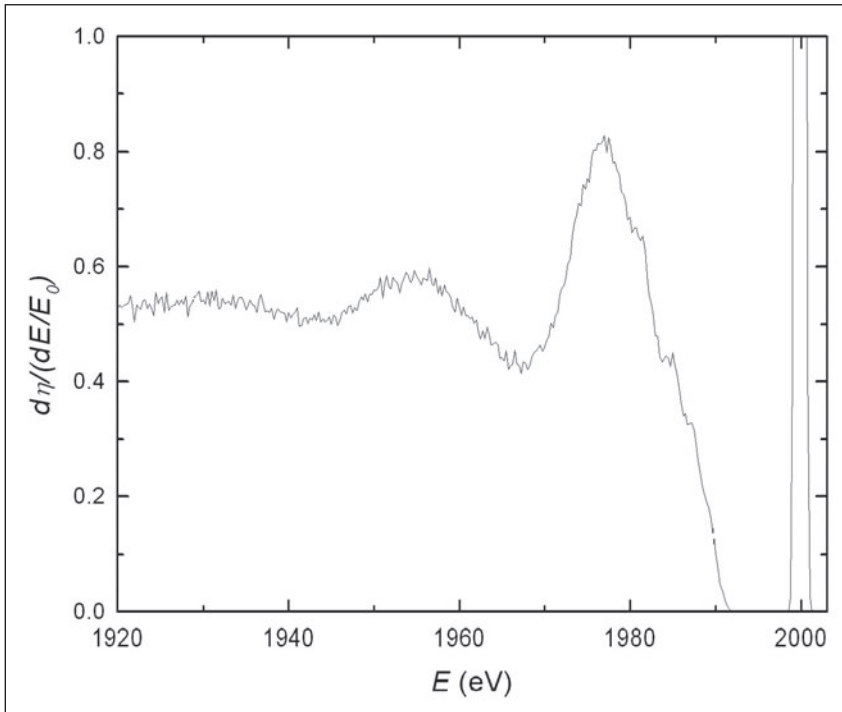


Fig. 1. Simulazione di Monte Carlo della distribuzione energetica $d\eta/d(E/E_0)$ degli elettroni retro-diffusi da un bersaglio di SiO_2 (diossido di silicio) in funzione dell'energia E degli stessi. L'energia iniziale E_0 degli elettroni è pari a 2000 eV.

troni incidenti e atomi del bersaglio. La figura evidenzia l'esistenza di un intervallo energetico tra il picco elastico ed il primo picco di perdita all'interno del quale non si contano elettroni retro-diffusi: questo significa che nessun elettrone viene retro-diffuso con un'energia compresa in quell'intervallo. In effetti, essendo il bersaglio un isolante, gli elettroni non possono cedere al solido energie inferiori al valore del *gap* energetico E_G tra banda di valenza e banda di conduzione e, di conseguenza, nessun elettrone del fascio primario può uscire dal materiale con un'energia compresa tra quella primaria sottratta di E_G e l'energia del picco elastico.

CONCLUSIONE

Si è descritto brevemente il metodo di Monte Carlo con l'obiettivo di mostrarne i limiti e le potenzialità: in particolare, si è evidenziata la

possibilità utilizzare questa tecnica matematica per risolvere complessi problemi di fisica relativi a processi che coinvolgono numeri molto grandi di particelle.

BIBLIOGRAFIA

- [1] DAPOR M., 2002 - L'intelligenza della vita, *Springer*, Milano.
- [2] DAPOR M., 2004 - Elogio del caso - *Atti Acc. Rov. Agiati*, a. 254, S. VIII, 4 (B): 37-46.
- [3] DAPOR M., 2005 - Electron-Beam Penetration in Surface Films - *Atti Acc. Rov. Agiati*, a. 255, S. VIII, 5 (B): 47-57.
- [4] DAPOR M., 2006 - Energy Loss Spectra of Low Primary Energy Electrons Back-scattered by Silicon Dioxide - *Surface Science*, 600: 4728-4734.
- [5] SOBOL I.M., 1984 - The Monte Carlo Method, *MIR*, Moscow.
- [6] DAPOR M., 2003 - Electron-Beam Interactions with Solids. Application of the Monte Carlo Method to Electron Scattering Problems - (Springer Tracts in Modern Physics), *Springer*, Heidelberg.
- [7] KOONIN S.E., MEREDITH D.C., 1990 - Computational Physics, *Addison-Wesley*, Redwood City.
- [8] PRESS W.H, TEUKOLSKY S. A, VETTERLING W.T., FLANNERY B.P., 1992 - Numerical Recipes in C. The Art of Scientific Computing - Second Edition, *Cambridge University Press*, Cambridge.
- [9] DAPOR M., ROPELE M., 2005 - Elaborazione dei dati sperimentali, *Springer*, Milano.

Indirizzo dell'autore:

Maurizio Dapor - Istituto Trentino di Cultura. Centro per la Ricerca Scientifica e Tecnologica, Via Sommarive, 18 - I-38050 Povo, Trento, Italia
