

# Integrazione numerica e metodo Monte-Carlo

Franco Bagnoli

October 14, 2019

## 1 Integrazione numerica

In molti casi dobbiamo calcolare numericamente degli integrali. La tecnica usata varia considerevolmente a seconda se di debba lavorare in una, due o comunque poche dimensioni e su un intervallo finito, o se si debbano eseguire degli integrali su un gran numero di variabili e/o su intervalli infiniti. Un altro elemento importante è se la funzione varia molto o se è abbastanza “liscia”.

Cominciamo con i casi più semplici, ovvero una funzione “liscia” da integrare su un intervallo finito, ovvero qualcosa del tipo

$$\int_a^b f(x)dx.$$

Può succedere che l'integrale non sia esprimibile con funzioni conosciute (o comunque semplici da calcolare), o magari che la funzione  $f$  stessa non sia conosciuta in forma analitica, ma sia il risultato di misure sperimentali. Per esempio, supponiamo che un accelerometro misuri l'accelerazione di un drone in volo ogni millesimo di secondo, e da questa serie di dati vogliamo derivare la posizione del drone sapendo il punto di partenza (e la sua velocità in tale punto). L'idea è quella di integrare l'accelerazione ottenendo la velocità, e poi integrare questa ottenendo la posizione.

Una tecnica è quella di approssimare la funzione usando dei polinomi, e poi integrare analiticamente la funzione interpolata. Però i polinomi di grado elevato tendono a coincidere con la funzione nei punti da cui li facciamo passare, ma a variare considerevolmente nei punti non campionati, per cui si preferisce usare questa approssimazione localmente, tra coppie o comunque un piccolo numero di punti.

Ci limiteremo qui al caso di funzioni campionate su punti equispaziati, perché questo è quello che di solito “esce” da un esperimento. In realtà, se si possono scegliere arbitrariamente i punti (intervalli di tempo) in cui campionare una funzione, è possibile avere la massima precisione dato il numero di punti utilizzato, per mezzo della quadratura di Gauss che non tratteremo.

Ma vediamo i metodi di integrazione

Il caso più semplice è ovviamente quello di tradurre la definizione “ingenua” dell'integrale, sommando le aree approssimate con dei rettangolini

$$\int_a^b f(x)dx = \lim_{N \rightarrow \infty} \sum_{i=0}^{N-1} f(x_i)\Delta x,$$

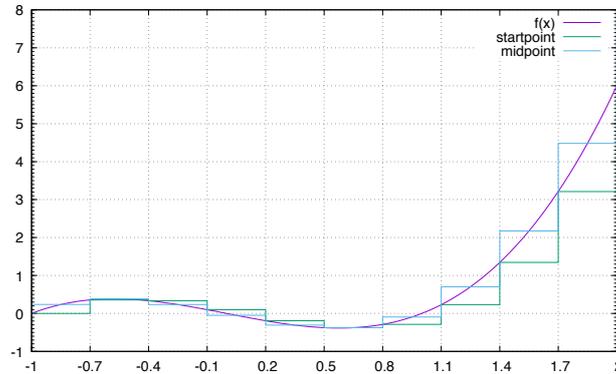


Figure 1: Integrazione con rettangoli calcolati nel punto iniziale o nel punto di mezzo.

dove  $\Delta x = (b - a)/N$  e  $x_i = a + i\Delta x$ .

questo metodo ha però il difetto di sovrastimare o sottostimare il valore dell'integrale a seconda del carattere crescente o decrescente della funzione, come si vede in Fig. 1. Si può ovviare a ciò calcolando la funzione nel punto di mezzo (stessa figura).

Si può migliorare la stima usando dei trapezi invece che dei rettangoli, che è equivalente ad usare la media dei valori della funzione nei vari estremi, o usando funzioni di grado maggiore, per esempio delle parabole o delle funzioni di terzo grado.

Cominciamo quindi a sviluppare il metodo usando il valore della funzione all'inizio e alla fine dell'intervallo. Come funzione test possiamo prendere una funzione facile da integrare, diciamo  $x^2$  tra 0 e uno, il cui integrale vale  $1/2$ .

Scriviamo un programma che dati  $N$  punti, campiona la funzione all'inizio, nel mezzo e alla fine di ogni intervallo. Chiamiamo  $a$  e  $b$  gli estremi dell'intervallo.

Il  $dx$  è sempre  $dx = (b/a)/N$ .

Nel primo caso abbiamo

$$I = \sum_{i=0}^{N-1} f(a + i \cdot dx) dx.$$

Nel secondo caso

$$I = \sum_{i=1}^N f(a + i \cdot dx) dx.$$

Nel terzo caso

$$I = \sum_{i=0}^{N-1} f(a + (i + 0.5) \cdot dx) dx.$$

Nel caso dei trapezi, invece

$$\begin{aligned}
 I &= \sum_{i=0}^{N-1} (f(a + i \cdot dx) + f(a + (i + 1)dx)) \frac{dx}{2} \\
 &= (f(a) + f(a + dx)) \frac{dx}{2} + (f(a + dx) + f(a + 2dx)) \frac{dx}{2} + \\
 &= f(a) \frac{dx}{2} + \sum_{i=0}^N f(a + i \cdot dx) dx + f(b) \frac{dx}{2}.
 \end{aligned}$$

Nel caso dell'interpolazione parabolica dobbiamo fare un po' di calcoli in più. Prendiamo tre punti consecutivi,  $x_0$ ,  $x_1$ ,  $x_2$ , corrispondenti a  $y_0 = f(x_0)$ ,  $y_1 = f(x_1)$ ,  $y_2 = f(x_2)$ , con  $x_0 = x_1 - dx$  e  $x_2 = x_1 + dx$ .

Troviamo la parabola che passa dal punto di mezzo  $(x_1, y_1)$ :

$$y = a(x - x_1)^2 + b(x - x_1) + y_1,$$

e imponiamo il passaggio per gli altri due punti

$$\begin{aligned}
 y_0 &= a(x_0 - x_1)^2 + b(x_0 - x_1) + y_1 \\
 y_2 &= a(x_2 - x_1)^2 + b(x_2 - x_1) + y_1.
 \end{aligned}$$

e con un po' di algebra si ottiene

$$\begin{aligned}
 a &= \frac{1}{2dx^2}(y_0 - 2y_1 + y_2) \\
 b &= \frac{1}{2dx}(y_2 - y_0).
 \end{aligned}$$

Calcoliamo l'integrale  $I_1$  corrispondente all'intervallo  $x_0 = x_2$ :

$$\begin{aligned}
 I_1 &= \int_{x_0}^{x_2} (a(x - x_1)^2 + b(x - x_1) + y_1) dx \\
 &= \left[ \frac{a(x - x_1)^3}{3} + \frac{b(x - x_1)^2}{2} + y_1 x \right]_{x=x_1-dx}^{x=x_1+dx} \\
 &= \frac{2}{3} a dx^3 + 2y_1 dx \\
 &= \frac{dx}{3} (y_0 + 4y_1 + y_2).
 \end{aligned}$$

Quindi l'integrazione parabolica (o con il metodo Cavalieri-Simpson è

```

dx = (b-a)/N;
tmp = f(a)*dx/3;
for (i=0; i<N; i+=2) {
    I += tmp;
    I += 4*f(a+(i+1)*dx)*dx/3;
    tmp = f(a+(i+2)*dx)*dx/3;
    I += tmp;
}

```

Le routine di integrazione numerica hanno una interfaccia del tipo

```
double integrate((double (* f)()), double x0, double t0, double deltat, int n);
```

dove `f(double x, double t)` è la funzione da integrare. Si noti che è stata dichiarata così per far capire al compilatore che si deve aspettare il puntatore ad una funzione che ritorna un `double`. Il resto degli argomenti sono il valore iniziale della  $x$  (zero per noi), il valore iniziale della  $t$  ( $a$  per noi), l'intervallo  $\Delta t = (b - a)/n$  e in numero di step.

## 2 Come passare parametri alle funzioni

Una possibilità è ovviamente quella di codificare i parametri nella definizione di funzione, ma questo non è sempre possibile né agevole. In particolare, quando dobbiamo fare integrali multidimensionali, gli integrandi dipendono da parametri che cambiano. Supponiamo di dover fare un integrale di una funzione di due variabili  $x$  e  $y$  che dipende da uno o più parametri  $a$  u

$$\int dx \int dy f(x, y; a)$$

che si può scrivere

$$\int dx F(x; a)$$

dove

$$F(x; a) = \int dy f(x; y; a)$$

Nel primo integrale la funzione integranda dipende da un parametro ( $a$ ), ma nel secondo da due ( $x$  e  $a$ ).

Per non dover riscrivere ogni volta le funzioni di integrazione dobbiamo trovare il modo di passare un numero variabile di parametri alle funzioni da integrare.

Per esempio: supponiamo di voler calcolare il potenziale del campo gravitazionale generato da un disco di materia (la "Terra piatta") o, equivalentemente, il campo elettrico generato da un disco di carica uniforme.

Usiamo coordinate cilindriche  $(\rho, \theta, z)$  (Figura 2). Chiamiamo il punto in cui vogliamo calcolare il campo (punto-campo)  $P$  e per simmetria basta calcolarlo sul piano  $\theta = 0$ , quindi indichiamo le sue coordinate con  $P = (x, 0, z)$ . Il potenziale è dato dalla sovrapposizione (somma) di quello generato da tutti i punti-sorgente  $Q$ , che stanno sul disco (supposto di spessore trascurabile) di raggio  $R$  sul piano  $z = 0$ . Le coordinate del punto sorgente sono dunque  $Q = (\rho, \theta, 0)$ .

Dato che supponiamo la carica (o massa) uniforme, l'unità di carica è  $dQ = \sigma dS$ , con  $dS = \rho d\rho d\theta$ . Ovviamente  $\rho$  varia tra 0 e  $R$  e  $\theta$  tra 0 e  $2\pi$ . Il potenziale di una carica  $dQ$  nel punto  $P$  è dato (a parte le costanti e il segno) da

$$d\psi = \frac{dQ}{|P - Q|} = \frac{dQ}{\sqrt{(x - \rho \cos(\theta))^2 + \rho^2 \sin^2(\theta) + z^2}} = \frac{dQ}{\sqrt{x^2 - 2x\rho \cos(\theta) + \rho^2 + z^2}}$$

Quindi il potenziale totale è

$$\psi(x, 0, z) = \int_0^R d\rho \int_0^{2\pi} d\theta \frac{\rho}{\sqrt{x^2 - 2x\rho \cos(\theta) + \rho^2 + z^2}}.$$

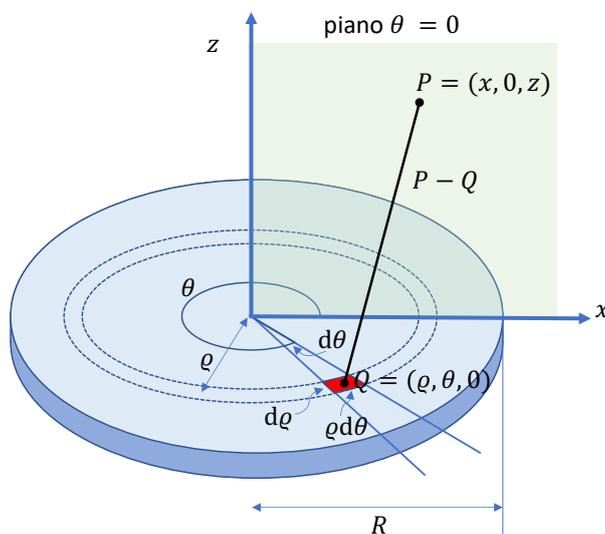


Figure 2: Coordinate usate per calcolare il potenziale della Terra piatta.

Questo integrale è difficile, si può risolvere con le funzioni elementari solo sull'asse ( $x = 0$ ),

$$\psi(0, 0, z) = \int_0^R d\rho \int_0^{2\pi} d\theta \frac{\rho}{\sqrt{\rho^2 + z^2}} = 2\pi \left( \sqrt{R^2 + z^2} - |z| \right),$$

che può servire da verifica.

Sempre per controllo, possiamo usare il calcolo (numerico) del potenziale fatto sull'asse  $x$  ( $z = 0, \theta = 0$ ), fig. 1.

Il sistema più semplice che si può usare per passare i parametri è quello di usare variabili globali,

```
double w; // parameter of f
double f(double x) {
    return cos(w*x);
}
double integrate(double (*f)(double), double a, double b, int N);
...

main() {
    w = 0.1;
    I = integrate(f, a, b, N);
}
```

che però non è molto elegante. Comunque per il problema della Terra piatta eccolo qui (per  $R = 1$ )

```
double x, z;
double rr; // tmp var
```

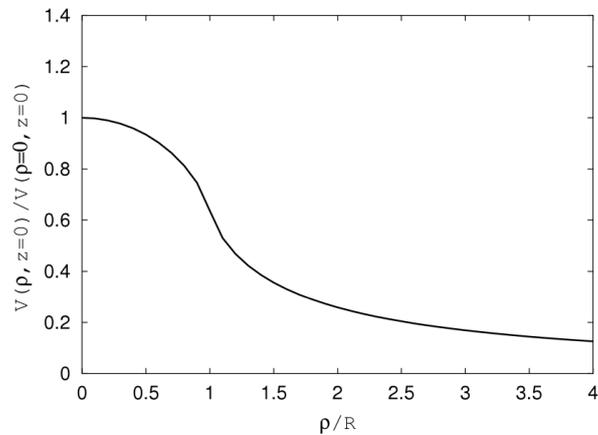


Figure 3: Figure from doi:10.1016/j.aml.2011.05.019

```

int N1;

double f1(double r) { // funzione integranda esterna
    rr=r; // passing r to f
    return integrate(f,0, 2*M_PI,N1); // this is the integral in \theta
}

double f(double t) { // funzione integranda interna
    return rr/sqrt(x*x-2*x*rr*cos(t)+rr*rr+z*z) ;
}

....
N = 100;
N1=100; // global vars
for (z=0.01;z<2; z+=0.05) { // z and x are passed through global vars
    for (x=0.1; x<2; x+=0.05) {
        I = integrate(f1, 0, 1, N) // this is the integral in r
    }
}

```

Una alternativa può essere quella di usare un vettore per passare i parametri alla funzione (così da non dover cambiare la definizione di `integrate` per passare 1, 2, ... parametri).

```

double integrate (double (*f)(double, double *),
                 double * params,
                 double a, double b, int N);

double f(double x, double * params);

main() {
    double params[10];

```

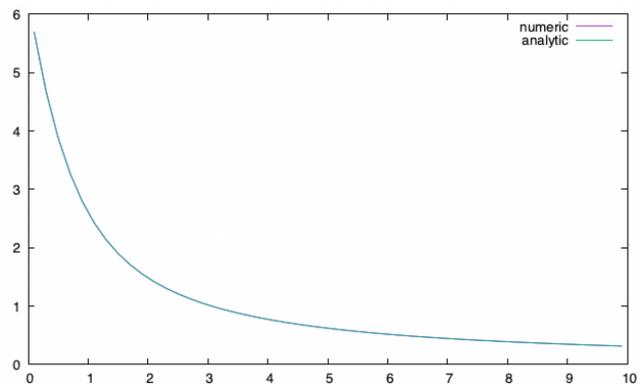


Figure 4: Potenziale (con segno invertito) sull'asse  $z$

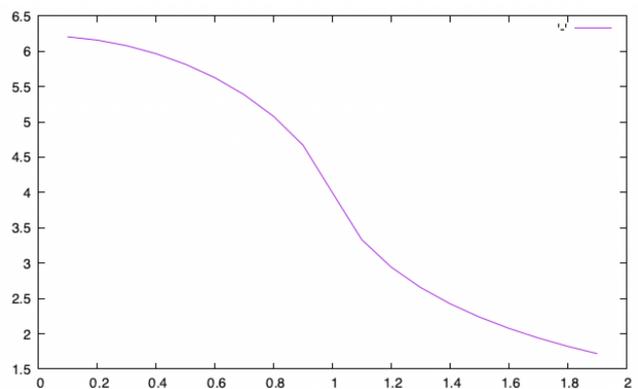


Figure 5: Potenziale (con segno invertito) sull'asse  $x$

```

params[0] = w;
I = integrate(f, params, a, b, N);
....
}

double integrate (double (*f)(double, double *),
                 double * params;
                 double a, double b, int N) {
    ...

    I = 0;
    dx = (b-a)/N;
    for (i=0; i<N; i++) {
        I += f(a+i*dx, params) * dx;
    }
    ...
}

```

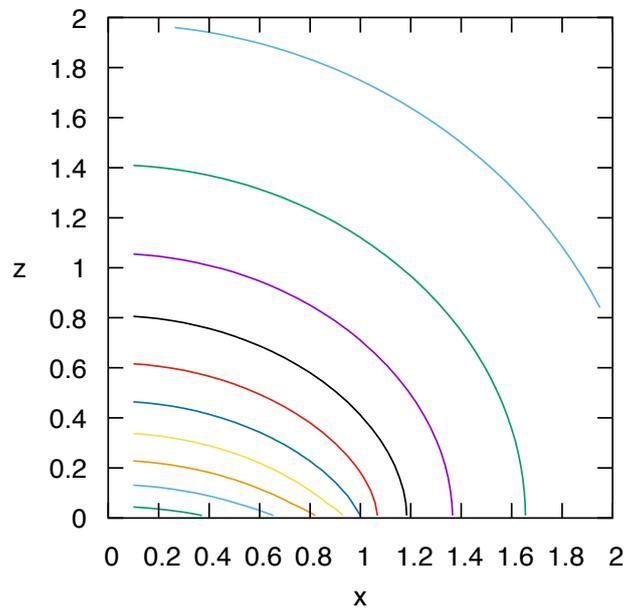


Figure 6: Linee equipotenziali per un disco uniformemente carico (Terra piatta).

per la Terra piatta abbiamo

```

double f(double t, double * params);
double f1(double r, double * params);

double integrate(double (*f)(double, double *), double * params,
    double a, double b, int N);

main() {
    double I;
    int N, N1;
    double params[10];
    double x, z;

    N = 100; // N dev'essere pari per integratep
    N1 = 100;
    params[0]=(double) N1;
    x = 0;
    params[1]=x;
    for (z=0.01; z<2; z+=0.05){
        for (x=0.1; x<2; x+=0.05) {
            params[1]=x;
            params[2]=z;
            I = integratet(f1, params, 0, 1, N);
        }
    }
}

```

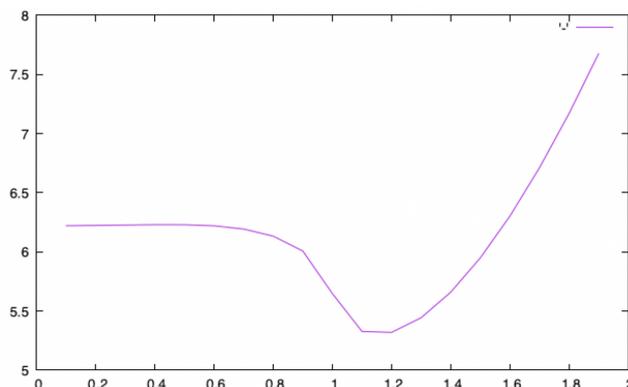


Figure 7: Potenziale (invertito) di una Terra piatta in rotazione con  $\omega = 1.8$ .

Il potenziale sull'asse  $z$  è riportato (invertito) in fig. 4, come si vede torna perfettamente con quello teorico. Il risultato sull'asse  $x$  è riportato in fig. 5, e coincide (almeno a prima vista) con quello dell'articolo di fig. 3.

Le linee di livello del potenziale sul piano  $xz$  sono riportate in fig. 6. Come si vede, anche sulla "terra" ( $z = 0, x < 1$ ) il potenziale non è piatto, quindi i laghi sono curvi in direzione radiale, e "piatti" lungo i cerchi  $\rho = \text{costante}$ .

Sono state disegnate con gnuplot:

```

set view map scale 1
unset surface
set contour base
set cntrparam levels auto 10 unsorted
set size square
unset key
set xlabel 'x'
set ylabel 'z' norotate
splot 'terrapiatta.dat' w l

```

Come esercizio, provate a "curare" il problema mettendo la Terra piatta in rotazione, ovvero aggiungendo un termine (sempre con il segno invertito)  $(1/2)\omega^2 x^2$  al potenziale. Per  $\omega = 1.8$  otteniamo un campo più o meno costante su quasi tutto il raggio. Ovviamente in questo caso si ha poi la forza di Coriolis...

### 3 Relazione con l'integrazione di equazioni differenziali

Si noti che se noi chiamiamo

$$y(x) = \int_a^x f(x') dx'$$

abbiamo che l'integrale  $y$  è la soluzione dell'equazione differenziale

$$\frac{dy}{dx} = f(x)$$

con la condizione al bordo  $f(a) = 0$ . Si possono quindi usare i metodi numerici per l'integrazione delle equazioni differenziali.

In generale una equazione differenziale del primo ordine si scrive

$$\frac{dx}{dt} = \dot{x} = f(x, t)$$

e in questo contesto l'integrale è dato dalla  $x$  e la variabile di integrazione dalla  $t$ , quindi la  $f$  dev'essere una funzione solo del tempo.

I metodi elencati sopra sono equivalenti a integrare le corrispondenti equazioni differenziali con passo fisso.

Si può verificare ciò scrivendo un programma che integri l'equazione differenziale

$$\dot{x} = t^\alpha$$

tra due estremi  $a$  e  $b$  in  $n$  step, usando vari metodi. Questo equivale a calcolare

$$x = \int_a^b t^\alpha dt = F(b) - F(a),$$

con

$$F(x) = \frac{1}{\alpha + 1} x^{\alpha+1}.$$

Nel listato `intnum.c` troviamo le routine per integrare secondo Eulero, Eulero ma calcolato nel punto di mezzo, Runge-Kutta del secondo e del quarto ordine.

Notate che Eulero è esattamente equivalente al calcolo con i rettangoli, Runge-Kutta del secondo ordine è equivalente al calcolo usando i trapezi, e Runge-Kutta del 4 ordine è come usare una funzione del terzo ordine. Si può verificare la precisione e la predizione per il Runge-Kutta del 4 ordine calcolando l'integrale di  $x^3$ : per quanto pochi intervalli si usino (anche uno solo), il Runge-Kutta del 4 ordine dà sempre il risultato esatto. Si noti anche che dato che si usano funzioni integrande che dipendono solo dal tempo, il Runge-Kutta del 4 ordine "spreca tempo" a calcolare due volte la stessa grandezza.

## 4 Integrazione stocastica

In molti casi, specialmente in meccanica statistica, dobbiamo eseguire integrali a molte dimensioni e/o su domini non limitati. In questo caso l'approccio di cui sopra è molto inefficiente: se partizioniamo il dominio (anche quando è limitato) in molti intervalli, abbiamo un numero astronomico di punti in cui dobbiamo valutare la funzione. Inoltre, non è chiaro in quali intervalli la funzione assume valori significativi, e in quali invece assume dei valori trascurabili.

Vediamo di sviluppare dei metodi stocastici, che usano i numeri casuali.

### 4.1 Direct sampling

Per prima cosa, poniamoci l'obiettivo di misurare la superficie del cerchio, supponendo di non conoscere la formula (oppure, conoscendo la formula, possiamo cercare un metodo per valutare numericamente  $\pi$ ).

L'idea è la seguente: racchiudiamo il nostro cerchio in un quadrato, non necessariamente tangente, e gettiamo casualmente tanti punti nel quadrato, contando quanti sono anche dentro il cerchio e quanti no.

Se la distribuzione dei punti è uniforme, il rapporto tra i punti che cadono nel cerchio rispetto a tutti i punti gettati è uguale (nel limite di numero di punti infinito) al rapporto tra l'area del cerchio e quella del quadrato.

Per generare i numeri casuali possiamo usare le routine che avete già utilizzato, oppure la funzione di libreria `drand48()` che usa un moltiplicatore congruenziale a 48 bit e fornisce già un numero `double` tra 0 e 1. Si inizializza con `srand48(time(NULL))`.

Prendiamo un quadrato  $[0, 1] \times [0, 1]$  e un quarto di cerchio di raggio  $r$  ( $r \leq 1$ ), tanto per fare le cose semplici. Generiamo due numeri  $x$  e  $y$  e sommiamo (nella variabile  $c$ ) tutti i casi per cui

$$x^2 + y^2 < r^2.$$

Il numero  $c/N$  approssimerà il rapporto tra l'area del quarto di cerchio,  $\pi r^2/4$  e quella del quadrato (uno). Per cui, se vogliamo stimare  $\pi$ , abbiamo

$$\pi = \lim_{N \rightarrow \infty} \frac{4c}{Nr^2}.$$

Il codice `pi.c` fa questo. Si noti che la convergenza è piuttosto lenta (tipicamente ordine  $1/\sqrt{N}$ , ma in compenso l'ordine di grandezza del risultato è quello giusto fin dall'inizio.

Come regola d'oro del Monte Carlo, il numero di punti che sono conteggiati (quelli nel cerchio in questo caso) rispetto a quelli totali dev'essere intorno alla metà. Se troppi punti cadono nel cerchio, o se ne cadono troppo pochi, la convergenza è più lenta. Si provi ad usare un raggio del cerchio molto piccolo per vedere l'effetto. Il raggio ottimale per la convergenza è intorno a 0.8.

## 4.2 Random walk sampling

Il metodo illustrato funziona finché il dominio è finito. Per prendere dimestichezza con il caso infinito, supponiamo di voler generare la distribuzione dei punti localmente, ovvero a generando un punto come variazione dell'ultimo punto utilizzato. È come se volessimo calcolare l'area di un cerchio molto grande (una pista di atterraggio per elicotteri) usando dei sassi lanciati a caso, ma senza riuscire a lanciare i sassi su tutta la pista.

L'idea è quella di usare un cammino casuale, ovvero dato un sasso lanciarne un altro a caso, in un certo range, e così via. Succederà prima o poi che un sasso cada fuori dal dominio. In questo caso dobbiamo restare fermi, lasciando cadere un altro sasso, come descritto da Werner Krauth nelle sue lezioni di introduzione al calcolo Monte Carlo disponibili su ArXiv (e allegate alla lezione). In questa maniera compensiamo il fatto che è più "difficile" entrare negli angoli del dominio, perché una volta entrati tutti i sassi che cadono fuori "contano" come se fossero caduti nell'angolo.

L'algoritmo non è molto diverso dal precedente, generiamo prima due numeri  $\tilde{x}$  e  $\tilde{y}$  variando  $x$  e  $y$  di una qualche quantità casuale (in un certo range), e se  $\tilde{x}$  e  $\tilde{y}$  stanno ancora nel nostro dominio, li usiamo per rimpiazzare  $x$  e  $y$ , che altrimenti mantengono i loro valori precedenti.

L'algoritmo è ovviamente più lento del precedente (dipende anche da quale valore usiamo come range). Si tenga d'occhio al solito l'acceptance rate.

### 4.3 Importance sampling

In molti casi vogliamo integrare una funzione su un dominio infinito, ovviamente la funzione va a zero all'infinito, ma non sappiamo in quale zona è decisamente diversa da zero (e quindi si concentra l'informazione).

Il calcolo Monte-Carlo è nato per effettuare integrali del tipo

$$\int_{-\infty}^{\infty} dx_1 \int_{-\infty}^{\infty} dx_2 \cdots \int_{-\infty}^{\infty} dx_n f(x_1, x_2, \dots, x_n) p(x_1, x_2, \dots, x_n)$$

dove  $p(x)$  è una distribuzione di probabilità tale che

$$\int_{-\infty}^{\infty} dx_1 \int_{-\infty}^{\infty} dx_2 \cdots \int_{-\infty}^{\infty} dx_n p(x_1, x_2, \dots, x_n) = 1,$$

ma noi ci limitiamo al caso monodimensionale.

L'idea del calcolo è quella di usare il random walk come prima, ma invece di fermarsi al bordo di un dominio ben definito, lasciare andare il walker dove vuole, ma indurlo a restare nella regione dove la probabilità è maggiore. Tutto questo, come vedremo, senza neppure calcolare a fondo la distribuzione di probabilità, perché per riuscire a normalizzarla (ovvero a far sì che il suo integrale valga uno) dovremmo anche in questo caso fare degli integrali che non sappiamo fare...

Supponiamo di avere un camminatore che fa una serie di passi da  $x = x(t)$  a  $y = x(t+1)$ , con una probabilità (da determinare)  $\tau(x \rightarrow y)$ . Vogliamo che l'insieme delle posizioni del camminatore, mediate su un tempo lungo, siano distribuite come la nostra  $p(x)$ . Una condizione sufficiente, detta del bilancio dettagliato, è che

$$p(x)\tau(x \rightarrow y) = p(y)\tau(y \rightarrow x)$$

ovvero che il "flusso" di probabilità portato dai camminatori si "bilanci" in media. Da qui otteniamo

$$\frac{\tau(x \rightarrow y)}{\tau(y \rightarrow x)} = \frac{p(y)}{p(x)}.$$

Ora, la nostra funzione di probabilità la si può scrivere

$$p(x) = \frac{1}{Z} g(x)$$

dove  $Z$  è una costante di normalizzazione (che non sappiamo calcolare) che fa sì che l'integrale sia uno.

Ma nella frazione la  $Z$  scompare, ovvero

$$\frac{\tau(x \rightarrow y)}{\tau(y \rightarrow x)} = \frac{g(y)}{g(x)},$$

quindi possiamo scegliere le  $\tau$  in maniera che siano sempre probabilità (maggiori di zero e minori di uno) ma con un certo grado di libertà. Questo è possibile se le  $g(x)$  sono sempre positive (o sempre negative, che è lo stesso).

Consideriamo per esempio il caso di voler integrare tra zero e infinito la funzione  $x \exp(-x^2)$ , che dà ovviamente  $\sqrt{\pi}/2$ .

Possiamo considerare  $g(x) = \exp(-x^2)$ , ottenendo

$$\frac{\tau(x \rightarrow y)}{\tau(y \rightarrow x)} = \frac{\exp(-y^2)}{\exp(-x^2)} = \exp(-(y^2 - x^2)).$$

La funzione  $\exp(-(y^2 - x^2))$  è sempre positiva, e minore di uno se  $y^2 > x^2$ , per cui possiamo prendere

$$\tau(x \rightarrow y) = \begin{cases} 1 & \text{se } y^2 > x^2, \\ \exp(-(y^2 - x^2)) & \text{altrimenti.} \end{cases}$$

Si può verificare che nella frazione di cui sopra una delle tau vale uno e l'altra prende il valore dell'esponenziale. Questa è la regola detta di Metropolis.

Nel caso specifico estendiamo la nostra funzione su tutto l'asse

$$\int_{-\infty}^{\infty} |x| p(x) dx$$

così non dobbiamo preoccuparci del bordo  $x = 0$ , e per la  $p(x)$  prendiamo

$$p(x) = \frac{1}{\sqrt{\pi}} \exp(-x^2).$$

L'integrale (che possiamo calcolare esplicitamente) vale

$$\int_{-\infty}^{\infty} |x| \frac{1}{\sqrt{\pi}} \exp(-x^2) = \frac{2}{\sqrt{\pi}} \int_0^{\infty} x \exp(-x^2) = \frac{1}{\sqrt{\pi}} (-\exp(-x^2))_0^{\infty} = \frac{1}{\sqrt{\pi}}$$

e quindi abbiamo un altro modo di calcolare  $\pi$ .

L'algoritmo è molto più semplice della spiegazione. Partiamo da  $x = 0$ . Come nel caso precedente generiamo una variazione (arbitraria) di  $x$  e applichiamo la regola della  $\tau$  di cui sopra: se  $y^2 < x^2$  accettiamo la mossa, altrimenti estraiamo un numero casuale  $r$  (`r=rand48()`) uniformemente distribuito tra 0 e 1. Si può verificare facilmente che  $r < p$  con probabilità  $p$ , e quindi confrontiamo  $r$  con  $\exp(-(y^2 - x^2))$ , accettando la mossa se è minore. In ogni caso accumuliamo la somma di  $|x|$ . Questa somma, divisa per il numero di passi, tende a  $1/\sqrt{\pi}$ , L'algoritmo corrispondente è nel file `pigaussian.c`.