
METODI STATISTICI E COMPUTAZIONALI

Stefania Spagnolo

Dipartimento di Matematica e Fisica, Univ. del Salento



METODI MONTE CARLO

Testi consigliati:

W. Kinzel & G. Reents "Physics by Computer" P. 157-163

W.R. Gibbs "Computation in Modern Physics" P. 25-46

[Dispense L. Angelini Università di Bari P. 29-44](#)

[Dispense D. Martello Università del Salento Cap. 1](#)

G. Cowan, "[Monte Carlo techniques](#)" in [The review of particle physics 2021](#)

LEZIONE 1

METODI MONTE CARLO

“A Monte Carlo technique is any technique making use of random numbers to solve a problem. (We assume for the moment that the reader understands what a random number is, although this is by no means a trivial point and will be treated later in some detail.)“

F. James **Rep. Prog. Phys.**, Vol. **43**, 1980

- Le tecniche Monte Carlo sono utilizzabili solo per affrontare problemi legati al calcolo delle probabilità ?
 - NO
 - Una tecnica Monte Carlo e' una qualunque tecnica che fa uso di numeri casuali per risolvere un problema.
- Il termine **simulazione Monte Carlo** fu coniato all'inizio della seconda guerra mondiale da Von Neumann e Ulam a **Los Alamos**, che studiavano la dinamica delle esplosioni nucleari nell'ambito del progetto Manhattan.
- Il nome fu ispirato alla casualità dei risultati nelle case da gioco.

METODI MONTE CARLO

- Storicamente, i primi calcoli su larga scala basati sul metodo Monte Carlo vennero eseguiti per lo studio di ***scattering e assorbimento di neutroni***. Questi processi hanno natura casuale, e dunque si può facilmente far corrispondere un campione ipotetico, costruito coi numeri casuali, al campione reale.
- D'altra parte ***i risultati ottenuti (valori medi, varianze etc) sono "deterministici"***, e potrebbero essere ottenuti con metodi tradizionali di calcolo analitico: in particolare, in questo caso si tratta del calcolo di integrali multidimensionali.
- Il metodo Monte Carlo può essere applicato ogni volta che si può stabilire una corrispondenza tra il risultato desiderato di un "calcolo" che si vuole effettuare o della stima di una grandezza e il comportamento atteso di un sistema stocastico.
- Tale corrispondenza può esistere per un sistema intrinsecamente probabilistico, ma anche per un sistema di natura deterministica.
- In ogni caso, l'appropriatezza del metodo al problema a cui è applicato dipenderà dalle proprietà matematiche del problema, non da una *somiglianza* tra il sistema stocastico e il problema oggetto di studio.

METODI MONTE CARLO

- ***Il metodo Monte Carlo è in sostanza un metodo di integrazione numerica***
- Le basi matematiche dell'integrazione Monte Carlo sono le basi della statistica
 - Concetti che ricorreranno:
 - Definizione di una variabile casuale
 - Distribuzioni delle variabili casuali: media, varianza, covarianza...
 - La legge dei grandi numeri
 - Il teorema del limite centrale
 - *Molti degli argomenti che toccheremo in questo corso sono quindi legati alle tecniche Monte Carlo*

METODI MONTE CARLO

In pratica

- Immaginiamo di essere interessati a un parametro (F) di una popolazione:
 - usiamo una sequenza di numeri casuali per costruire un campione della popolazione, ed otteniamo delle stime statistiche del parametro.
 - F può essere un numero reale, o un array, o semplicemente una variabile logica.
- La stima ottenuta col metodo Monte Carlo dipenderà *anche* dall'insieme dei numeri casuali utilizzati nel calcolo.
- **Usare una tecnica Monte Carlo, quindi, significa usare una sequenza di numeri casuali come uno strumento per calcolare una grandezza.**
- **Esempio**
 - Ipotizziamo che \mathbf{X} sia una variabile casuale con valore di aspettazione \mathbf{A} .
 - Utilizzare una tecnica Monte Carlo per determinare A significa generare n numeri secondo la distribuzione di probabilità a cui obbedisce la variabile casuale \mathbf{X} (x_1, \dots, x_n) e stimare A come la media campionaria \hat{A}_n

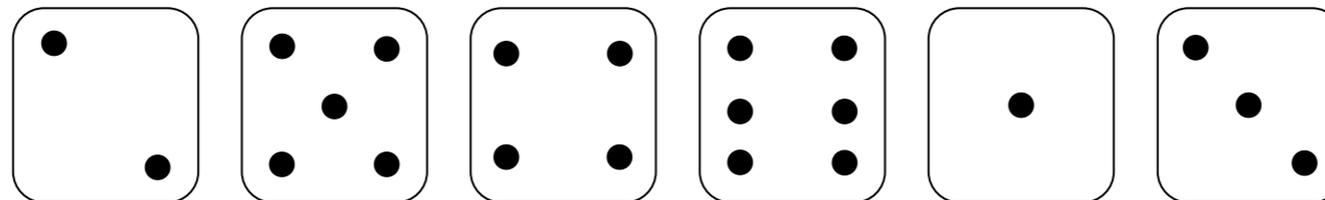
$$A \sim \hat{A}_n = \frac{1}{n} \sum_{i=1}^n X_i$$

METODI MONTE CARLO

- Per la legge dei grandi numeri (nella formulazione forte)
 - $\hat{A}_n \rightarrow A$ quando $n \rightarrow$ infinito
- In generale, se il programma è ben scritto, *la successione di valori casuali della variabile aleatoria X cambia ogni volta che si esegue nuovamente il codice, ma la grandezza A da stimare non cambia.*
- Vediamo quindi la **differenza tra tecnica Monte Carlo e Simulazione:**
 - La simulazione consiste nella generazione di numeri casuali fine a se stessi. Si parla di simulazione, cioè, quando lo scopo non è quello di stimare una grandezza (A nel caso precedente) ma quando è l'insieme di numeri casuali stesso lo scopo ultimo del processo.
 - La tecnica Monte Carlo invece serve a stimare valori deterministici.

METODI MONTE CARLO

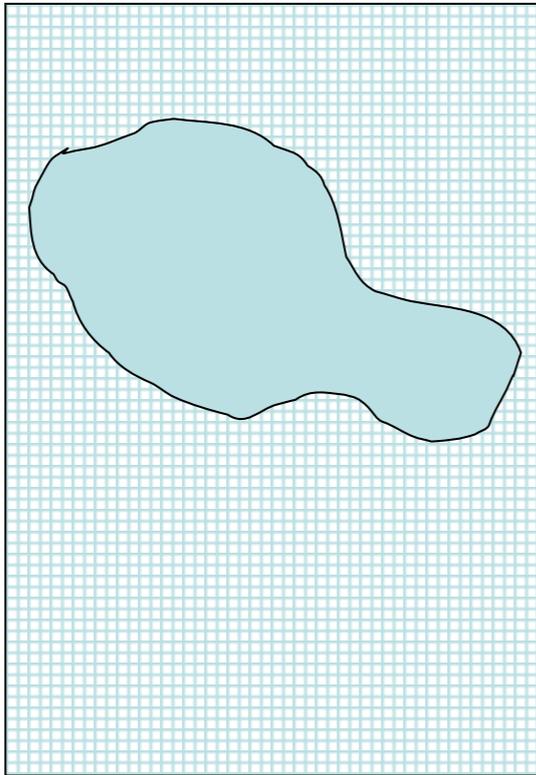
- Un esempio concreto di simulazione:



- in questo caso ciò che conta è la successione casuale di numeri casuali in se stessa; la domanda “Qual è il prossimo numero ?” ha una risposta **casuale**
 - => **Simulazione**
- la simulazione diventa Monte Carlo nel momento in cui uso le estrazioni casuali dei numeri da 1 a 6 per determinare un parametro, per esempio la media, della distribuzione di probabilità dei numeri ottenuti al lancio del dato
 - => Simulazione **Monte Carlo**

METODI MONTE CARLO

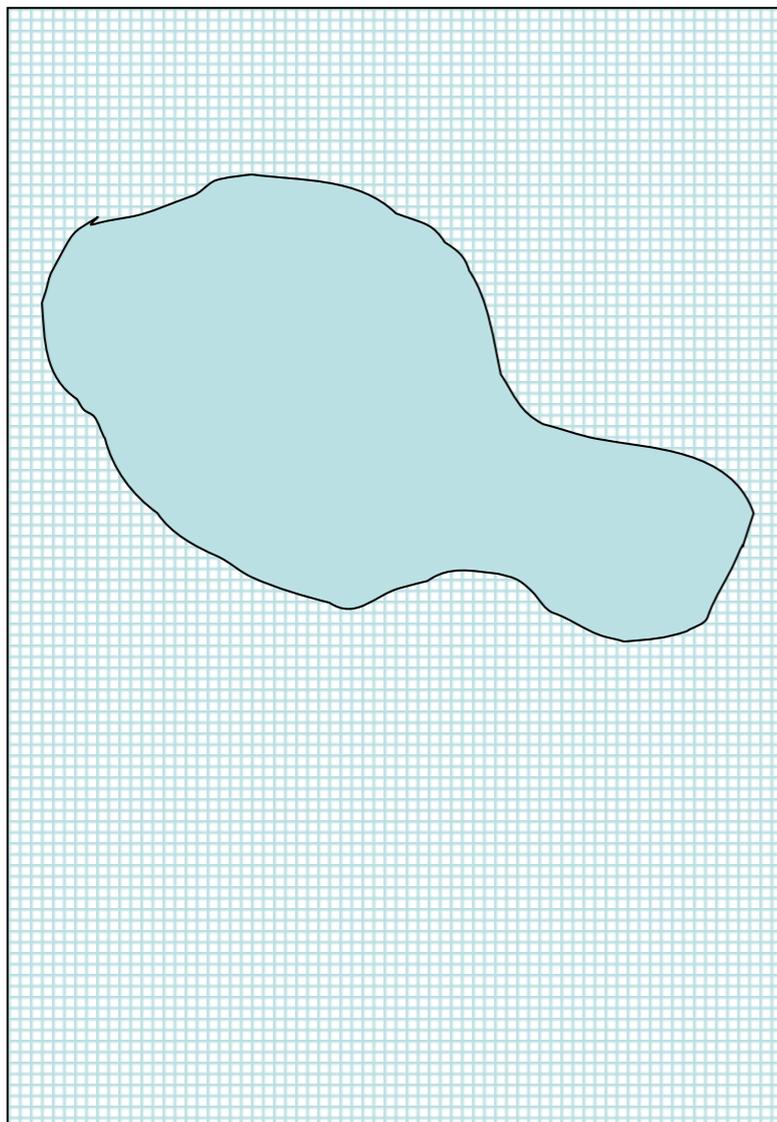
- Misurare l'area della regione azzurra



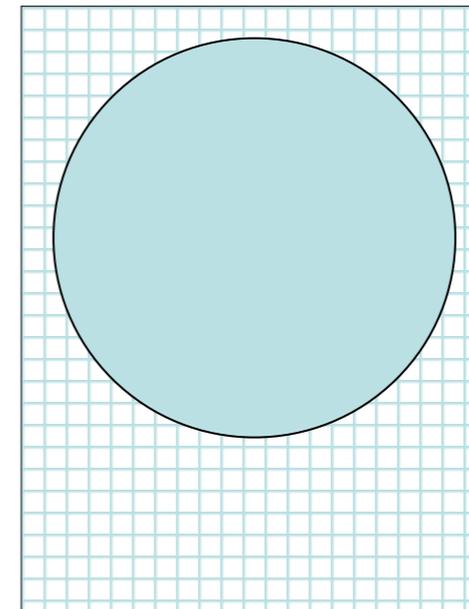
- Non ovvio qual sia il risultato

METODI MONTE CARLO

- Misurare l'area della regione azzurra



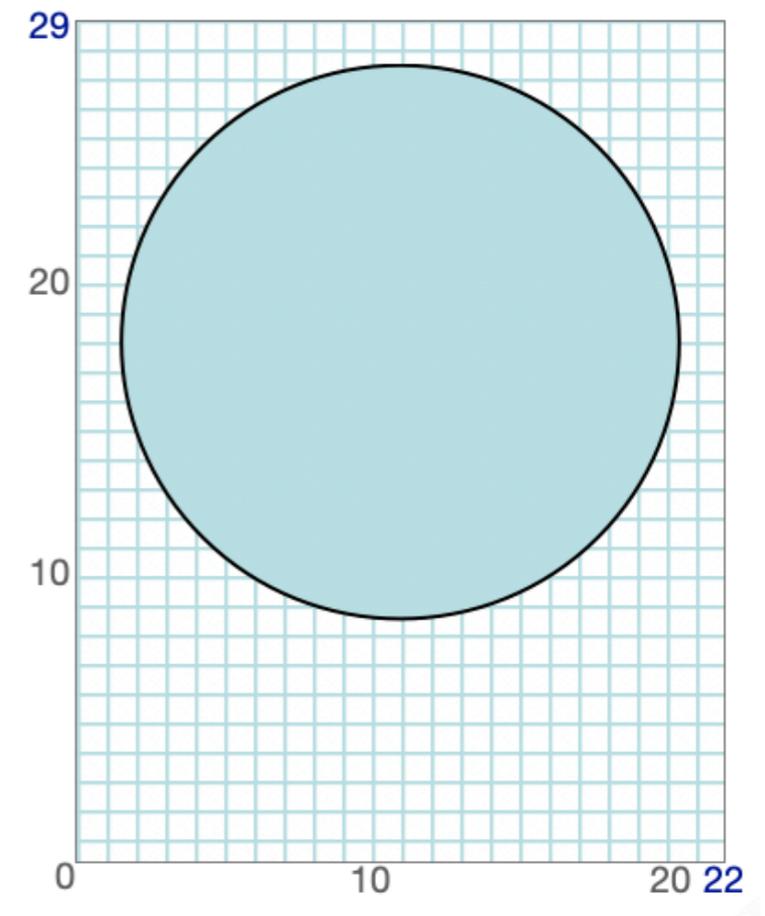
- Partiamo da un caso dal risultato noto: Misurare l'area del cerchio azzurro



- Voglio usare una successione di numeri casuali (tecnica Monte Carlo) per stimare un parametro (la superficie del cerchio azzurro) che ho disegnato sul mio foglio.

METODI MONTE CARLO

- Partiamo da un caso il cui risultato è noto: Misurare l'area del cerchio azzurro
 - Estraiamo N punti nel rettangolo di dimensioni 22 in x e 29 in y
 - = usiamo una sequenza di numeri casuali per x_i e y_i
 - Estraiamo x_i tra 0 e 22, secondo una distribuzione uniforme
 - Estraiamo y_i tra 0 e 29, secondo una distribuzione uniforme
 - Stabiliamo se (x_i, y_i) è interno al cerchio
 - Valutiamo $f = N_{interni}/N$
 - **Area del cerchio = $f \times$ Area del rettangolo**



METODI MONTE CARLO

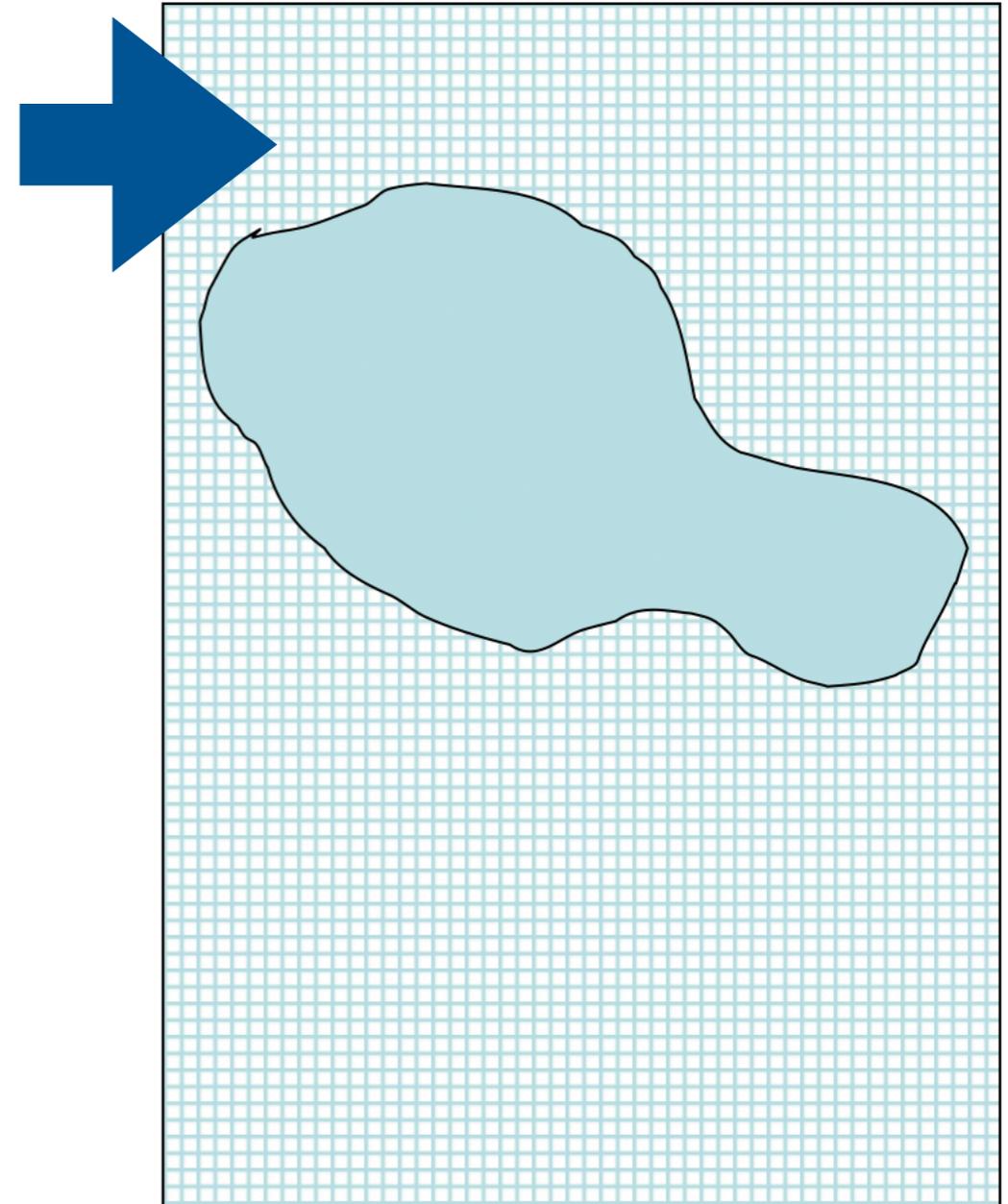
Può sembrare poco efficiente?

Forse, ma *in alcuni casi e' l'unica tecnica disponibile*

e in generale, come vedremo ha il considerevole vantaggio di permetterci di stimare a priori l'errore che si commette nella valutazione del parametro.

L'utilizzo di una tecnica Monte Carlo non dipende dalla natura stocastica di un certo problema, ma dalla capacità che abbiamo di usare una sequenza di numeri casuali per risolvere un certo problema.

Notiamo anche che alla fine la tecnica Monte Carlo ci ha permesso di calcolare un integrale. ***In generale i Metodi Monte Carlo sono metodi di integrazione.***



GENERAZIONE DI NUMERI CASUALI AL CALCOLATORE

Come sono generati ?

- I computers svolgono i calcoli in maniera deterministica ! => **numeri pseudo-casuali**
- I numeri generati al computer hanno un numero finito di cifre, **k**. Una variabile intera potrà essere scritta nella forma
 - $x = a_0 2^0 + a_1 2^1 + a_2 2^2 + a_3 2^3 + \dots$ con a_i uguale a 0 (oppure 1) con probabilità 1/2
- Una variabile così costruita (con $i=0, \dots, k$) ammetterà 2^k valori diversi possibili. **Se è possibile estrarre in maniera casuale** i valori degli a_i ,
 - x avrà una distribuzione uniforme sui 2^k valori discreti;
 - $x/(2^k - 1)$ sarà distribuita tra 0 e 1.

# cifre k	Numero binario				# valori rappresentabili =	2^k
1	a_0				2	2^1
2	$a_1 a_0$				4	2^2
3	$a_2 a_1 a_0$				8	2^3
4	$a_3 a_2 a_1 a_0$				16	2^4

$X = 0, 1, 2, 3, 4, 5, 6, 7$
 $y = x / (2^3 - 1) \rightarrow$ compreso tra 0 e 1



GENERAZIONE DI NUMERI CASUALI AL CALCOLATORE

- **Se è possibile estrarre in maniera casuale...**
 - **E' possibile, ma non banale, attraverso regole algoritmiche deterministiche**
 - In generale un calcolatore puo' produrre sequenze di numeri pseudo-casuali

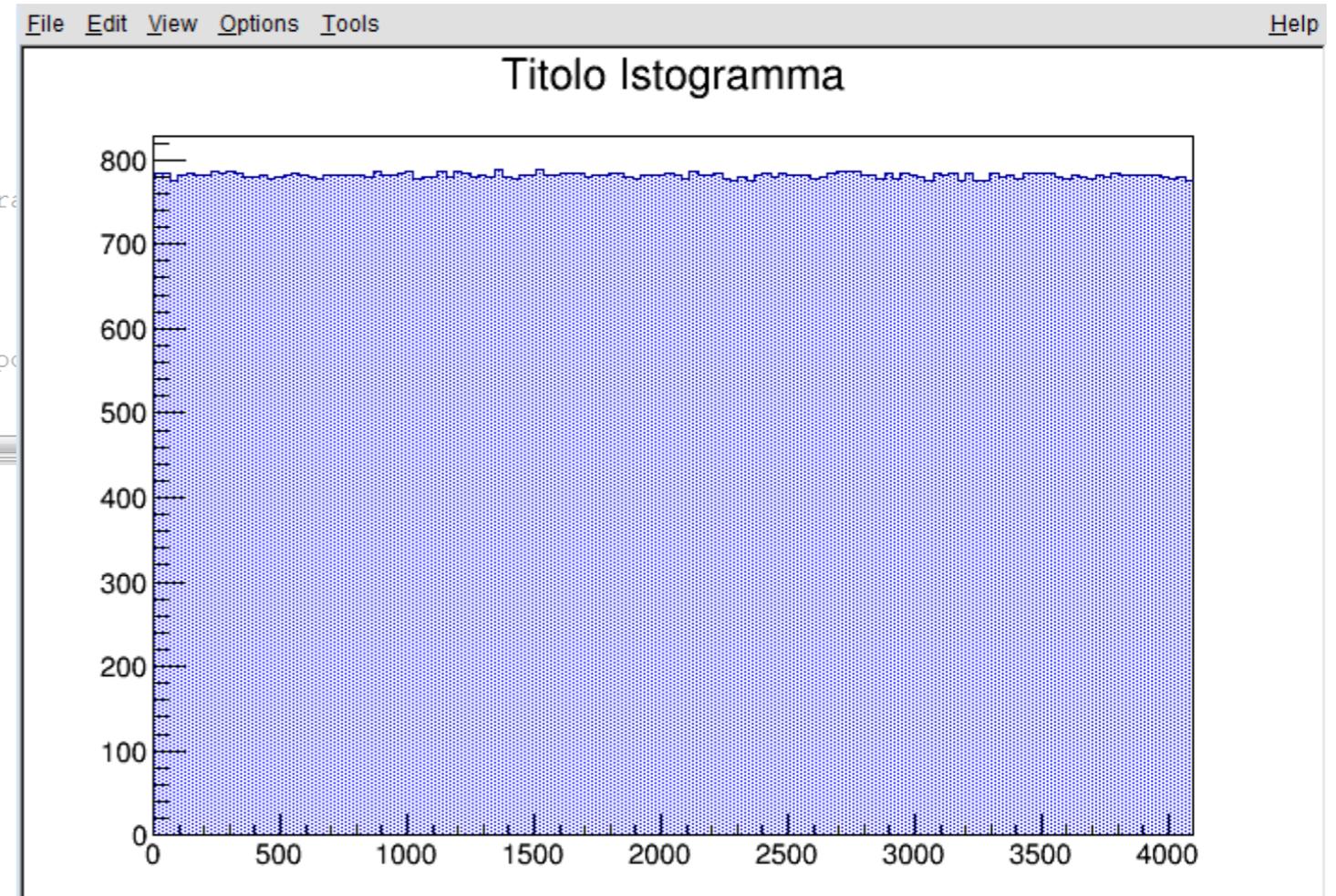
- Ogni numero (pseudo) casuale estratto da un computer sarà *funzione* dei numeri estratti in precedenza. La sequenza è riproducibile se si parte dallo stesso valore iniziale (**seme**) e parametri (**regola "algoritmica" di generazione**)
- Una possibile *regola* e' data da: $x_n = \text{mod} (\mathbf{a} x_{n-1} + \mathbf{c}, \mathbf{m})$ **a, c, m parametri**
 - **Generatore Congruente Lineare o di Lehmer** $X_0 = \text{seme}$

GENERATORE LINEARE CONGRUENTE

```

4 // Includo le librerie degli "oggetti" che servono
5 #include "TRandom.h"
6 #include "TH1D.h"
7 #include "TFile.h"
8 // Creo una script di nome Generatore
9 void Generatore() {
10     int N=100000;
11     // creo un oggetto file lo collego al file di nome esempiol.root e gli dico di crearlo (ricrearlo) quindi è un output
12     TFile f("esempiol.root","RECREATE");
13     // creo un oggetto histogramma
14     TH1D * h = new TH1D ("h","Titolo Istogramma",128,0,4096);
15     // definisco i parametri del generatore
16     int a=5;
17     int c=13;
18     int k=4096;
19     int seme=123456;
20     // inizializzo la sequenza
21     int x=seme;
22     for (int i=0; i<N ; i++) {
23         // generatore lineare
24         x=(x*a+c)%k;
25         // Riempio l'oggetto histogramma con i numeri random genera
26         h->Fill(x);
27     }
28     // scrivo l'oggetto histogramma nel file
29     h->Write();
30     // chiudo il file attraverso il metodo dell'oggetto di tipo
31     f.Close();
32 }

```



GeneratoreUnifRandLCG.C

GENERATORE LINEARE CONGRUENTE

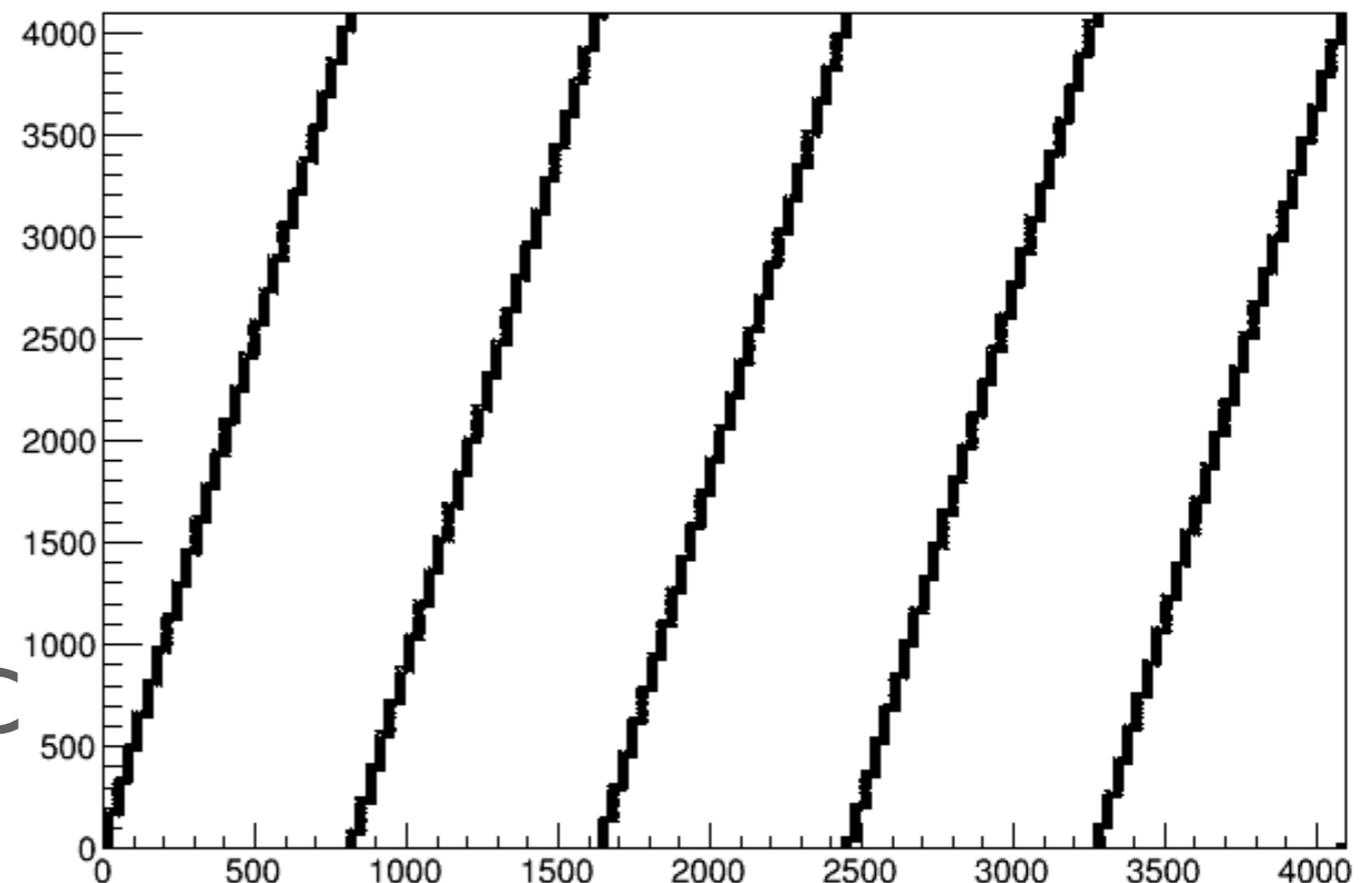
```

4 // Includo le librerie degli "oggetti" che mi servono
5 #include "TRandom.h"
6 #include "TH1D.h"
7 #include "TFile.h"
8 // Creo una script di nome Generatore
9 void Generatore(){
10     int N=100000;
11     // creo un oggetto file lo collego al file di nome esempio1.root e gli dico di crearlo (ricrearlo) quindi è un output
12     TFile f("esempio1.root","RECREATE");
13     // creo un oggetto histogramma
14     TH2D * h = new TH2D ("h","Titolo Istogramma",128,0,4096,128,0,4096);
15     // definisco i parametri del generatore
16     int a=5;
17     int c=13;
18     int k=4096;
19     int seme=123456;
20     // inizializzo la sequenza
21     int r=seme;
22     int x,y;
23     for (int i=0; i<N ; i++) {
24         // Generatore lineare
25         r=(r*a+c)%k;
26         x=r;
27         r=(r*a+c)%k;
28         y=r;
29         // Riempio l'oggetto histogramma con i numeri random
30         h->Fill(x,y);
31     }
32     // scrivo l'oggetto histogramma nel file
33     h->Write();
34     // chiudo il file attraverso il metodo dell'oggetto
35     f.Close();
36 }

```

File Edit View Options Tools

Titolo Istogramma



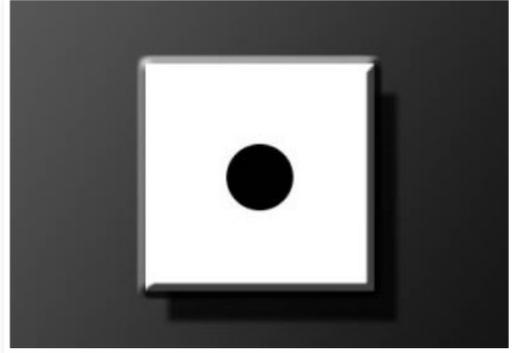
GeneratoreUnifRandLCG.C

UN ESEMPIO DI SIMULAZIONE: LANCIO DEL DADO

```

1 //
2 // esempio di utilizzo del generatore di numeri random in root
3 //
4 // Includo le librerie degli "oggetti" che mi servono
5 #include "TRandom.h"
6 #include "TH1D.h"
7 #include "TFile.h"
8 // Creo una script di nome Dado
9 void Dado() {
10 // creo un "oggetto" generatore di numeri random
11   TRandom R(12345);
12   int N=100000;
13 // creo un oggetto file lo collego al file di nome esempio1.root e gli dico di crearlo (ricrearlo) quindi è un output
14   TFile f("dado.root", "RECREATE");
15 // creo un oggetto histogramma
16   TH1D * h = new TH1D ("h", "Titolo Istogramma", 40, 0, 6.1);
17 // definisco i parametri del generatore
18   for (int i=0; i<N ; i++) {
19     double x;
20 // Uso l'oggetto generatore di numeri random per generare nume
21     x=R.Uniform();
22 // Converto in numeri interi compresi tra 1 e 6
23     int d=x*6+1;
24 // Riempio l'oggetto histogramma con i numeri random generati
25     h->Fill(d);
26   }
27 // scrivo l'oggetto histogramma nel file
28   h->Write();
29 // chiudo il file attraverso il metodo dell'oggetto di tipo fi
30   f.Close();
31 }
32

```

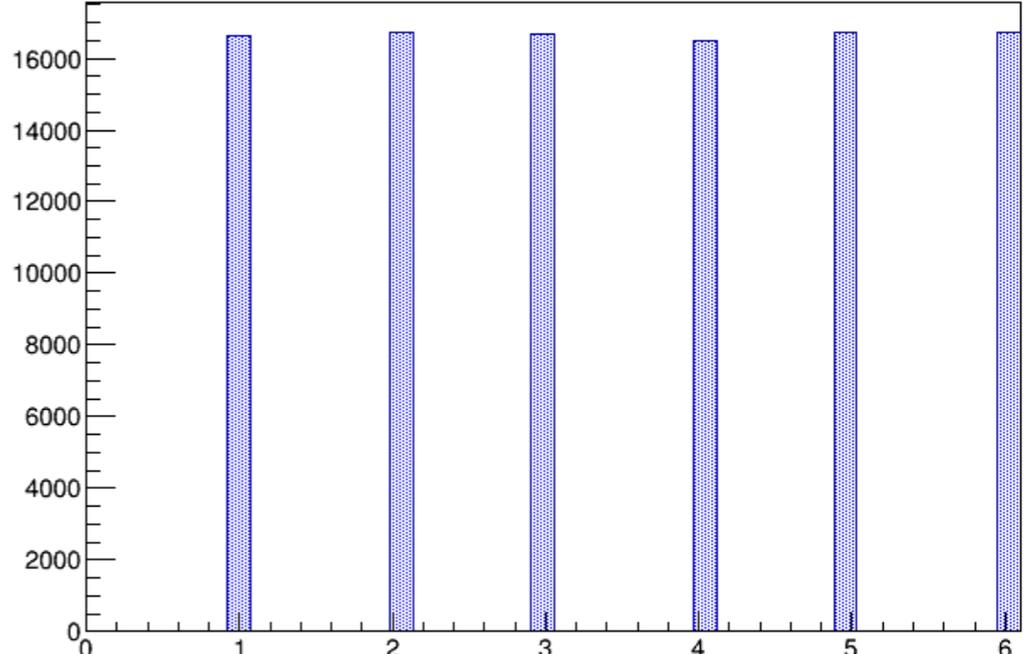


```

File Edit View Options Tools

```

Titolo Istogramma

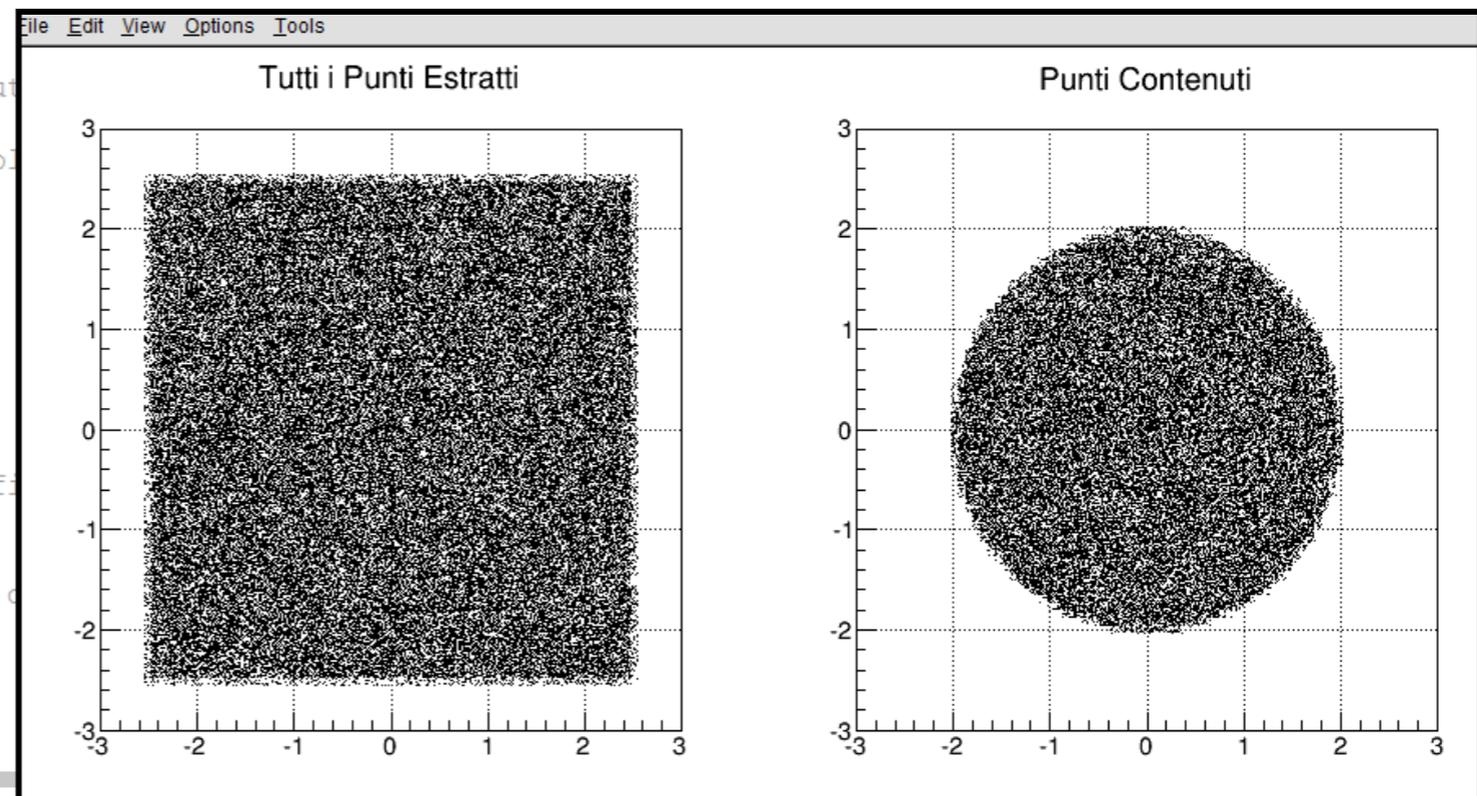
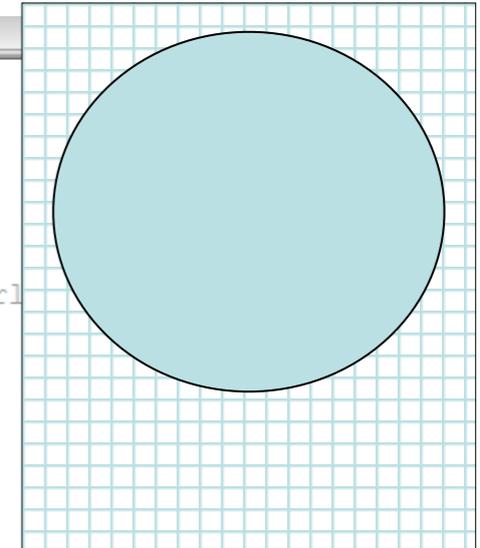


UN ESEMPIO DI MONTE CARLO: L'AREA DEL CERCHIO

```

7  #include "TFile.h"
8  // Creo una script di nome Cerchio
9  void Cerchio() {
10 // creo un "oggetto" generatore di numeri random
11   TRandom R(344545);
12   int N=100000;
13 // creo un oggetto file lo collego al file di nome esempi1.root e gli dico di crearlo (ricreare)
14   TFile f("cerchio.root","RECREATE");
15 // Raggio del cerchio
16   double raggio=2.;
17 // creo un oggetto histogramma
18   TH2D * h1 = new TH2D ("h1","Tutti i Punti Estratti",80,-3.,3.,80,-3.,3.);
19   TH2D * h2 = new TH2D ("h2","Punti Contenuti",80,-3.,3.,80,-3.,3.);
20 // inizializzo sequenza di conteggio
21   int In=0;
22   for (int i=0; i<N ; i++) {
23     double x,y;
24     // Uso l'oggetto generatore per generare punti in maniera uniforme
25     // su un quadrato di lato 5 centrato in 0,0
26     x=R.Uniform()*5.-2.5;
27     y=R.Uniform()*5.-2.5;
28     // Riempio l'oggetto histogramma con tutti i punti
29     h1->Fill(x,y);
30     // Riempio l'oggetto histogramma con solo i punti contenuti nel cerchio
31     if (sqrt(x*x+y*y)<raggio) {
32       h2->Fill(x,y);
33       In++;
34     }
35   }
36 // Calcolo area
37   double area=5.*5.*In/N;
38   printf("Area cerchio %lf\n",area);
39 // scrivo gli oggetti histogramma nel file
40   h1->Write();
41   h2->Write();
42 // chiudo il file attraverso il metodo Close()
43   f.Close();
44 }

```



ESERCIZI

- Da http://www.dmf.unisalento.it/~spagnolo/MSC_aa21_22/IntroduzioneROOT/ Esempi/
 - macroCollection.C
 - macro1(), macro2(), makeHist()
- Da http://www.dmf.unisalento.it/~spagnolo/MSC_aa21_22/MetodiMC/Esempi/
 - GeneratoreUnifRandRoot.C
 - Dado.C
 - Carte.C
 - GeneratoreUnifRandLCG.C

FINE
