

---

# METODI STATISTICI E COMPUTAZIONALI

---

Stefania Spagnolo

*Dipartimento di Matematica e Fisica, Univ. del Salento*



---

# APPLICAZIONI DELLA TECNICA 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](#)

---

# APPLICAZIONI DELLA TECNICA MONTE CARLO

---

## ***Metodi MC per l'integrazione di funzioni***

- rigetto
- Media di  $[b-a]f(x)$

## ***Metodo MC per implementare un generatore di numeri pseudocasuali non uniforme***

- rigetto
- inversione

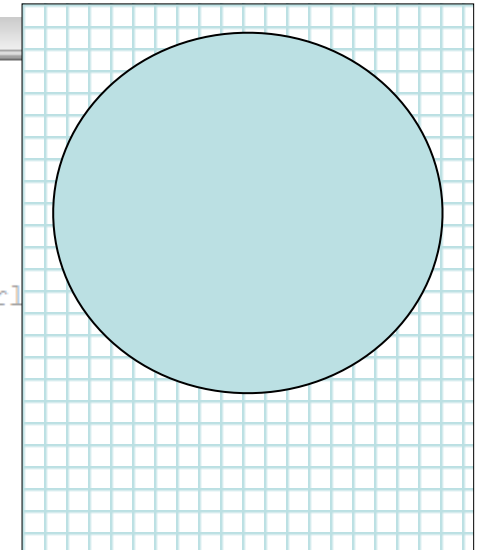
---

# LEZIONE 2

---

# UN ESEMPIO DI MONTE CARLO: L'AREA DEL CERCHIO

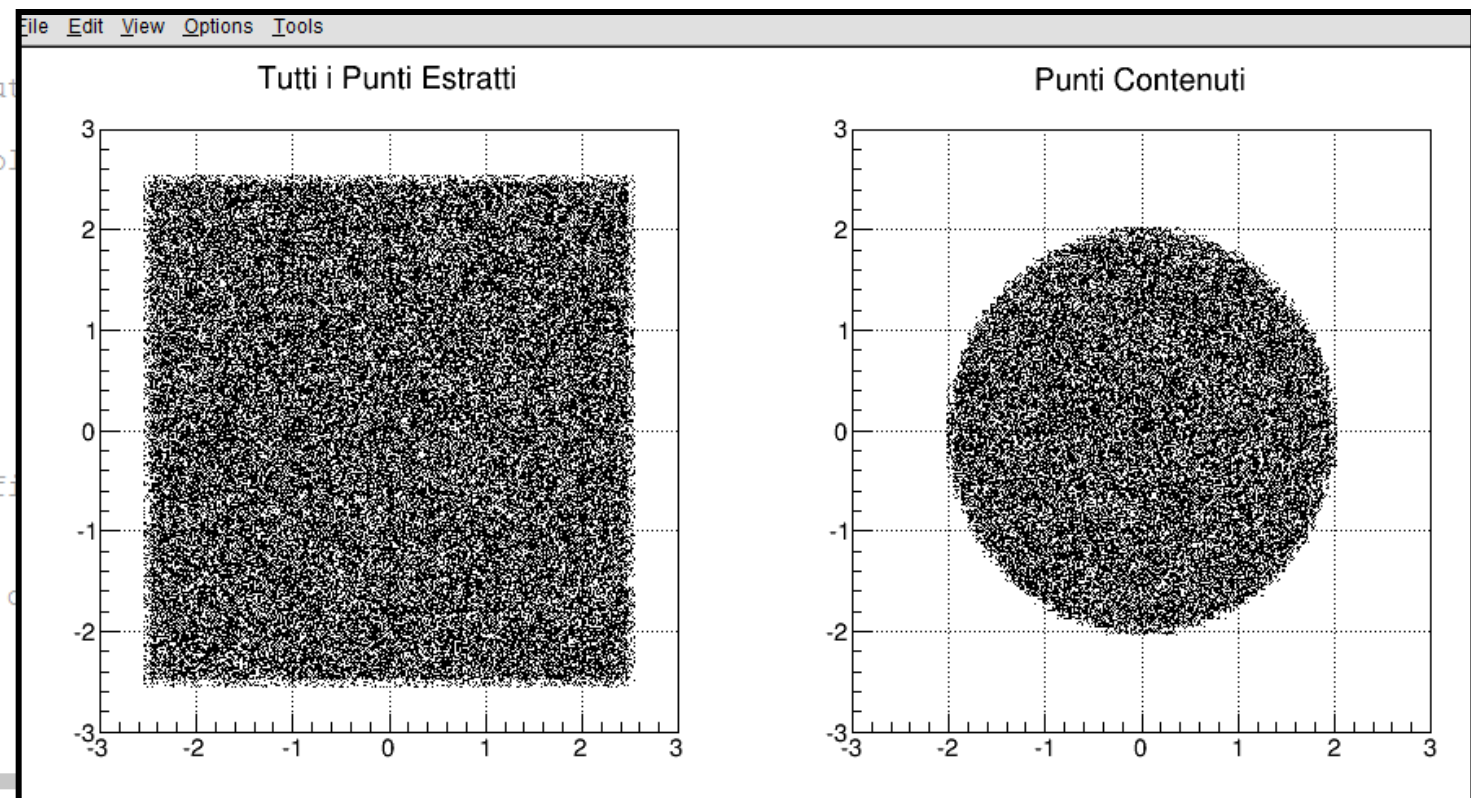
## Macro: Cerchio.C



```

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 }

```



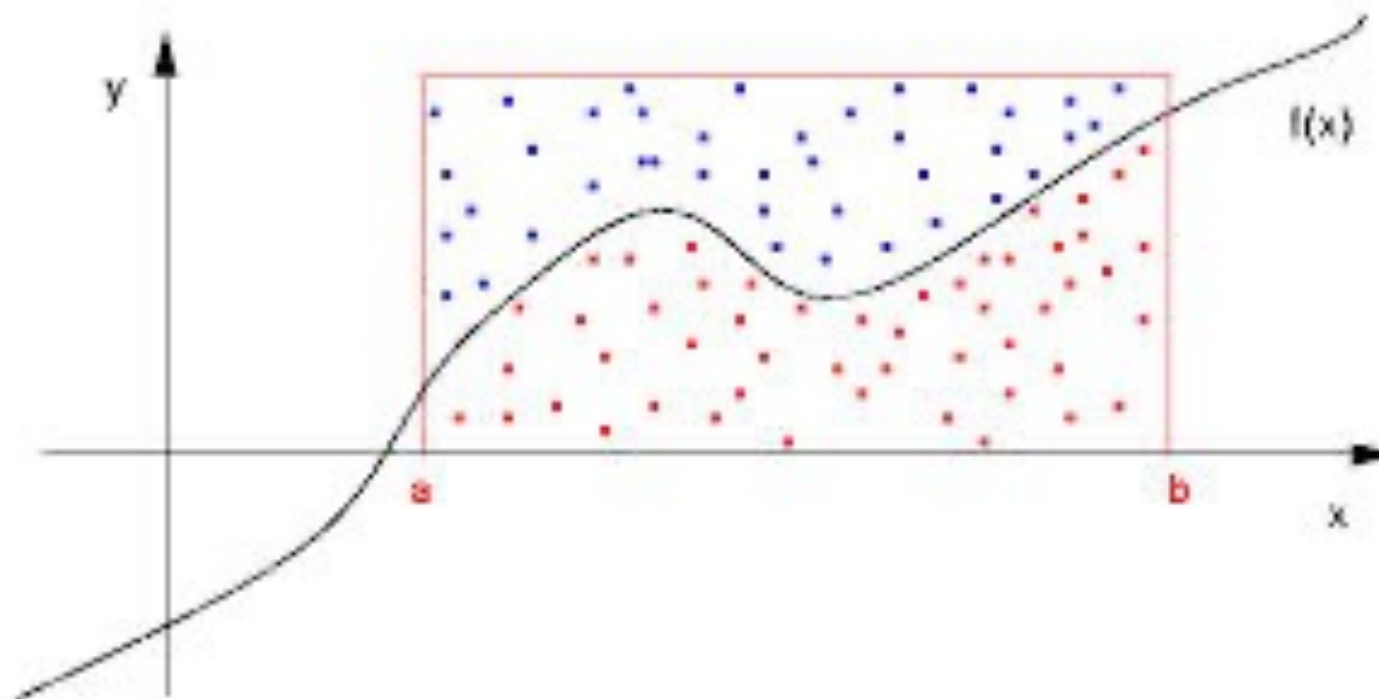
# MISURA DI INTEGRALI CON IL METODO DELLA RECEZIONE O RIGETTO

- La tecnica Monte Carlo che abbiamo usato per determinare la superficie di un cerchio può essere usata per calcolare un integrale

Calcoliamo l'integrale

$$\int_1^{10} (2x^2 + 1) dx = 675$$

$$= \left[ \frac{2}{3}x^3 + x \right]_1^{10} = 2000/3 + 10 - 2/3 - 1$$



# MISURA DI INTEGRALI CON IL METODO DELLA REIEZIONE O RIGETTO

- **Primo passo:** identificare un rettangolo (di area  $A$ ) che inscrive la funzione nell'intervallo di integrazione.
- **Secondo passo:** estrarre  $N$  punti casuali all'interno del rettangolo e contare il numero  $N_a$  di punti (accettati) che cadono sotto la curva che rappresenta la funzione
- **Terzo passo:** la frazione di punti accettati sul totale moltiplicata per  $A$  da il valore dell' integrale

```

hitormiss.C x FiguraIntegrale.C Cerchio.C
//
// calcoliamo l'integrale di questa funzione
double f(double x) {
    return 2.*x*x+1.;
}
void Reiezione() {
    // creo un "oggetto" generatore di numeri random
    TRandom R(12345);
    int N=10000;
    // creo un oggetto file lo collego al file di nome Reiezione.root
    // e gli dico di crearlo (ricrearlo) quindi è un output
    TFile f("Reiezione.root","RECREATE");
    TH2D * h1 = new TH2D ("h1","Tutti i Punti Estratti",80,0.,11.,80,0.,210);
    TH2D * h2 = new TH2D ("h2","Punti Contenuti",80,0.,11.,80,0.,210);
    // inizializzo sequenza di conteggio
    int In=0;
    for (int i=0; i<N ; i++) {
        double x,y;
        // Uso l'oggetto generatore di numeri random per generare numeri random in maniera uniforme
        x=R.Uniform()*9+1;
        y=R.Uniform()*201;
        // Riempio l'oggetto histogramma con tutti numeri random generati
        h1->Fill(x,y);
        // Riempio l'oggetto histogramma con solo i punti contenuti
        if (y<f(x)) {
            h2->Fill(x,y);
            In++;
        }
    }
    // Calcolo area
    double area=9.*201.*In/N;
    printf("Integrale = %lf\n",area);
    // scrivo gli oggetti histogramma nel file
    h1->Write();
    h2->Write();
    // chiudo il file attraverso il metodo dell'oggetto di tipo file.
    f.Close();
}

```



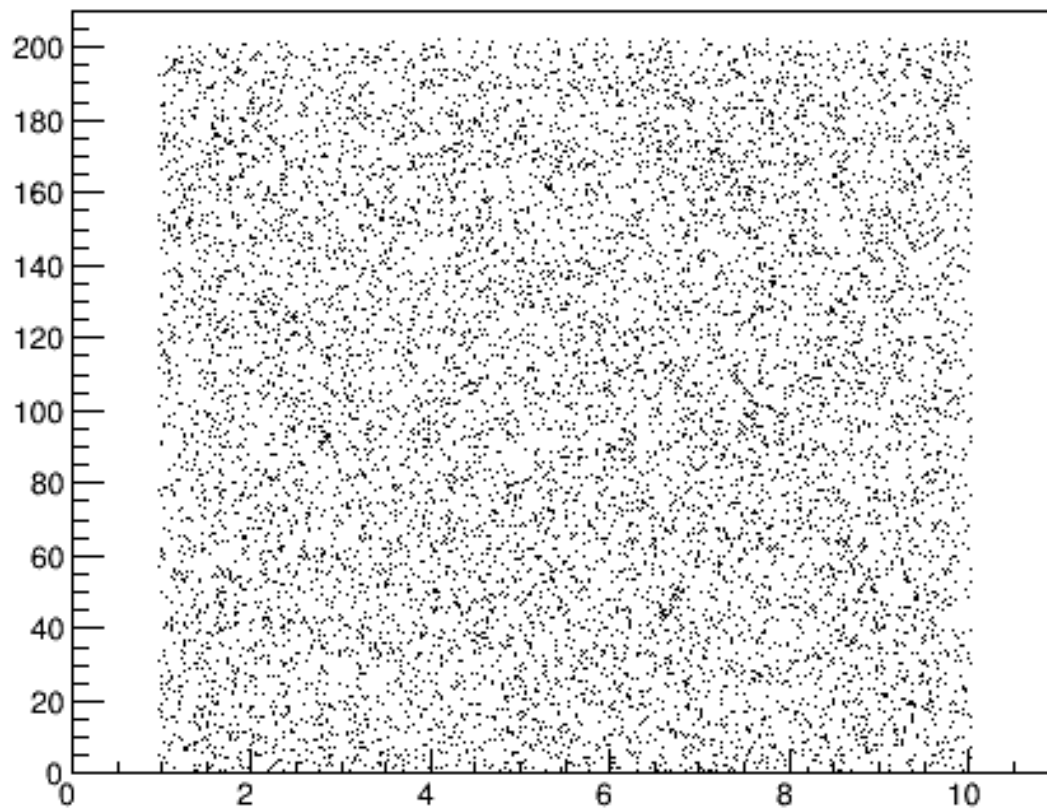
# MISURA DI INTEGRALI CON IL METODO DELLA REIEZIONE O RIGETTO

- Quale rettangolo ?
- Mappa dei punti estratti casualmente
- Mappa dei punti accettati

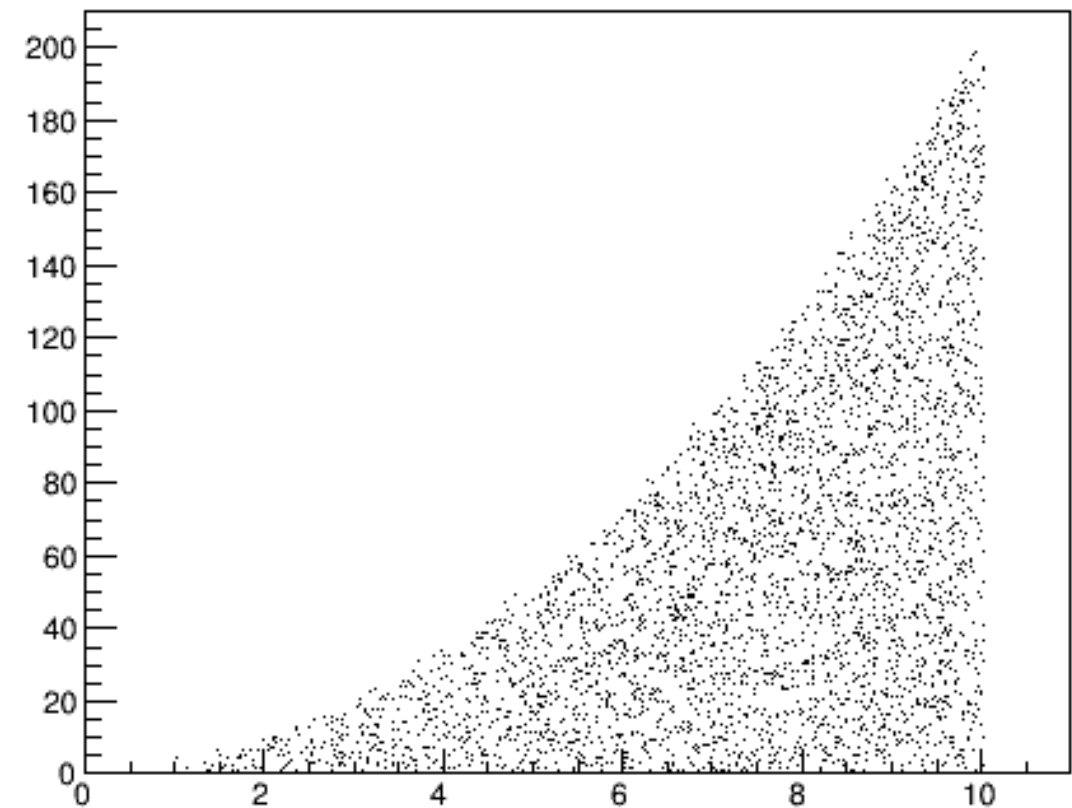
File Edit View Options Tools

Help

Tutti i Punti Estratti



Punti Contenuti



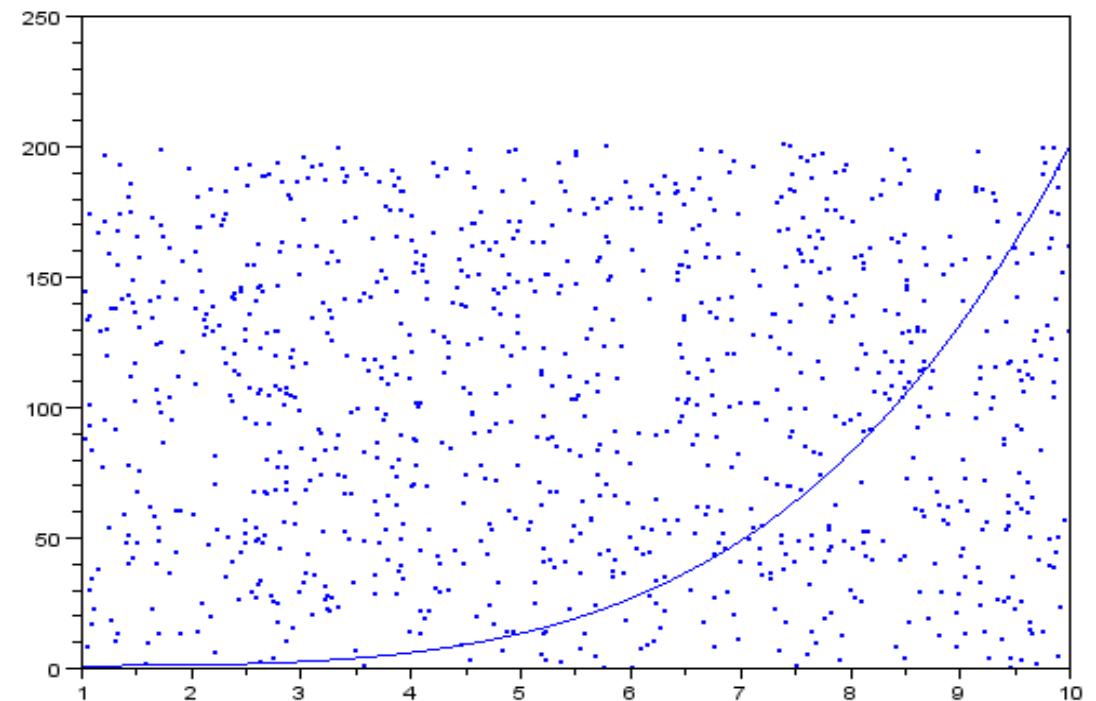
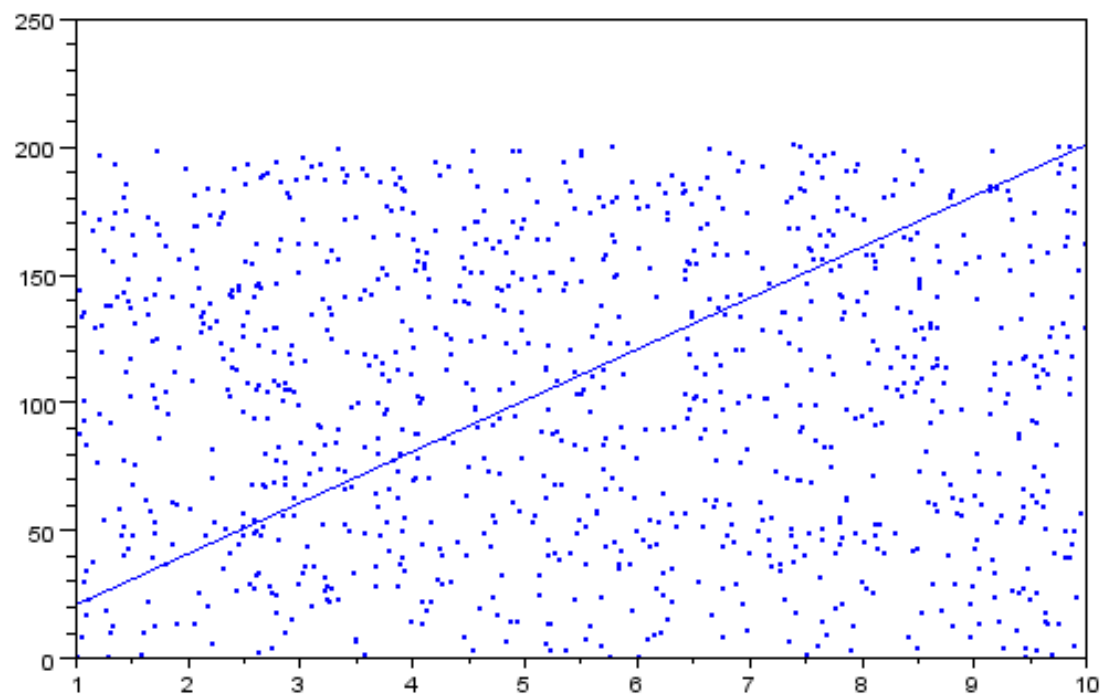


# MISURA DI INTEGRALI CON IL METODO DELLA RECEZIONE O RIGETTO

- Con il metodo del rigetto è possibile calcolare integrali definiti anche di funzioni  $f(x)$  non definite positive, basta, infatti, considerare la funzione:  
 $g(x) = f(x) - f_{min} > 0$  sempre nel dominio di integrazione se  $f_{min}$  e' il minimo di  $f(x)$ 
  - $$\int_a^b f(x)dx = \int_a^b g(x)dx + f_{min}(b - a)$$
- Non e' possibile calcolare con questo metodo integrali di funzioni con singolarità anche se matematicamente (analiticamente) integrabili
  - Solo per funzioni limitate nel dominio di definizione e' possibile determinare il minimo e il massimo, necessari alla costruzione del rettangolo

# MISURA DI INTEGRALI CON IL METODO DELLA REIEZIONE O RIGETTO

- Efficienza del metodo:
  - NOTA:
    - Numerosità' del campione che contribuisce alla stima statistica dell'integrale e'  $N_{accettati}$
    - efficienza  $\rightarrow \epsilon = N_{accettati}/N_{totali}$
    - numero di estrazioni totali  $N_{estrazioni-totali} = 2N$
- E' evidente che il metodo è tanto più efficiente quanto più l'area del rettangolo che include la funzione è prossima alla superficie che si deve determinare.



# MISURA DI INTEGRALI CON IL METODO DELLA REIEZIONE O RIGETTO

## Macro: Integrale\_Reiezione.C

- Efficienza del metodo:
  - NOTA:
    - Numerosità' del campione che contribuisce alla stima statistica dell'integrale e'  $N_{accettati}$
    - efficienza  $\rightarrow \epsilon = N_{accettati}/N_{totali}$
    - numero di estrazioni totali  $N_{estrazioni-totali} = 2N$
- E' evidente che il metodo è tanto più efficiente quanto più l'area del rettangolo che include la funzione è prossima alla superficie che si deve determinare.
- Stima dell'integrale
  - $I_{MC,rejection} = Area \times \frac{N_{acc}}{N} = Area \times \epsilon$
  - Qual e' l'errore statistico ?
    - Vedremo che l'errore statistico sulla nostra stima di epsilon e'  $\sigma(\epsilon) = \sqrt{\epsilon(1 - \epsilon)/N}$  (distribuzione di prob. binomiale)
    - Quindi l'errore su  $I_{MC,rejection} = Area \times \sqrt{\epsilon(1 - \epsilon)/N}$

# 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
  - ***Definizione di una variabile casuale***
  - ***Distribuzioni delle variabili casuali. Media, varianza, covarianza...***
  - ***La legge dei grandi numeri***
  - ***Il teorema del limite centrale***

torniamo su questo concetto

---

# INTEGRAZIONE CON IL METODO MONTE CARLO

---

# LA LEGGE DEI GRANDI NUMERI

- La legge dei grandi numeri mi garantisce di poter calcolare un integrale con tecniche Monte Carlo.
- In termini generici, **per la legge dei grandi numeri** si può dire che:
  - la media di una sequenza di variabili aleatorie indipendenti e caratterizzate dalla stessa distribuzione di probabilità (n misure della stessa grandezza, n lanci della stessa moneta ecc.) è una approssimazione, che migliora al crescere di n, della media della distribuzione

Variabile aleatoria	Distribuzione di probabilità	Media (o valore di aspettazione)
$x$	$g(x)$	$\mu = \int_a^b g(x)x dx$
$y = f(x)$	$g(x)$	$\mu = \int_a^b f(x)g(x) dx$



# INTEGRAZIONE DI FUNZIONE CON IL METODO MC

Supponiamo di voler valutare il seguente integrale

$$I = \int_a^b f(x) dx \quad (2.2)$$

dove  $f(x)$  è una funzione reale di variabile reale, ovviamente integrabile in  $[a, b]$ , con  $b > a$ . Consideriamo la seguente funzione

$$g(x) = \begin{cases} \frac{1}{b-a} & \text{per } a \leq x \leq b, \\ 0 & \text{altrove.} \end{cases}$$

Densità di probabilità  
uniforme tra  $a$  e  $b$  della  
variabile casuale  $x$

L'integrale può essere riscritto nella forma

$$I = (b - a) \int_a^b f(x) g(x) dx$$

$\langle F \rangle$

Valore di aspettazione  $\mu = \langle F \rangle$   
di  $f(x)$  per  $x$  variabile casuale  
distribuita con probabilità  
uniforme tra  $a$  e  $b$

$g(x)$  rappresenta una distribuzione di probabilità uniforme in  $[a, b]$ ; possiamo quindi considerare l'integrale come valore di aspettazione di  $f(x)$ , con  $x$  variabile aleatoria distribuita secondo  $g(x)$ .

# INTEGRAZIONE DI FUNZIONE CON IL METODO MC

- Per la legge dei grandi numeri, estratti  $N$  valori di  $x$ ,  $\{x_i\}$ , la media campionaria (aritmetica) di  $f(x_i)$   $\bar{F}$  tende a  $\langle F \rangle$  per  $N \rightarrow \infty$

Media sul campione di  $N$  valori di  $x$

$$\bar{F} = \frac{1}{N} \sum_{i=1}^N f(x_i) \xrightarrow{N \rightarrow \infty} \langle F \rangle$$

- L'integrale  $I$  sull'intervallo  $[a, b]$  di una funzione  $f(x)$  può essere stimato come  $(b - a)\bar{F}$  con approssimazione tanto maggiore quanto più grande è  $N$

- Qual è l'errore che commetto ?

$$= \int_a^b f(x)g(x)dx$$

$$= \frac{1}{b - a} \int_a^b f(x)dx$$

Integrale di  $f(x)$

$$= \frac{1}{b - a} I$$

# IL TEOREMA DEL LIMITE CENTRALE

- **La somma di variabili aleatorie indipendenti e identicamente distribuite** con media  $\mu$  e varianza  $\sigma^2$ , **indipendentemente dalla distribuzione di probabilità** di partenza, al tendere della dimensione del campione a infinito tende a distribuirsi come una **variabile casuale gaussiana**

$$\bar{F} = \frac{1}{N} \sum_{i=1}^N f(x_i) \rightarrow \langle F \rangle$$

$\bar{F}$  si distribuisce secondo una pdf gaussiana con media  $\mu = \langle F \rangle$  e varianza  $\text{var}(\bar{F}) = \sigma^2 = \text{var}(f) / N$

La deviazione standard  $\sigma$  di  $\bar{F}$  sarà  $\sqrt{\text{var}(f)} / \sqrt{N} = \sigma(f) / \sqrt{N}$

# INTEGRAZIONE DI FUNZIONE CON IL METODO MC

Lo stimatore MC dell'integrale  $I$ , cioè  

$$I_{MC} = (b - a)\bar{F}$$
, è affetto da un errore pari a  

$$(b - a) \sigma(f(x)) / \sqrt{N}$$

che possiamo stimare dal campione  
 come  $(b-a) \text{RMS}(\{f(x_i)\}) / \text{sqrt}(N)$

Stima campionaria della deviazione standard di  $f$

- **L'integrale  $I$  sull'intervallo  $[a,b]$  di una funzione  $f(x)$  può essere stimato come  $(b - a)\bar{F}$  con approssimazione tanto maggiore quanto più grande è  $N$** 
  - *Qual è l'errore che commetto ?*

# INTEGRAZIONE DI FUNZIONE CON IL METODO MC

- ESEMPIO

Calcoliamo l'integrale  $\int_1^{10} (2x^2 + 1) dx = 675$

- Per le considerazioni appena illustrate (da legge dei grandi numeri e teorema del limite centrale) ho

$$\frac{1}{N} \sum_{i=1}^N (2x_i^2 + 1) \xrightarrow{N \rightarrow \infty} \frac{1}{b-a} \int_a^b (2x^2 + 1) dx$$

$x_i$  valori della variabile casuale  $x$  distribuita in maniera uniforme tra  $a$  e  $b$

- Quindi se estraggo  $N$  numeri casuali  $\{x_i\}$  calcolo la  $f(x_i)$  e ne faccio la media ottengo asintoticamente il valore dell'integrale diviso l'ampiezza dell'intervallo.
- Nel nostro esempio:  $675/9=75$



# INTEGRAZIONE DI FUNZIONE CON IL METODO MC

## Macro: Integrale\_MonteCarlo.C

```

8 //
9 // Calcolo integrale con Metodo Monte Carlo
10 //
11 // calcoliamo l'integrale di questa funzione
12 double f(double x) {
13     return 2.*x*x+1.;
14 }
15 void Integrale() {
16     // creo un "oggetto" generatore di numeri random
17     TRandom R(345345);
18     int N=100000;
19     // Inizializzo somma e somma2
20     double somma=0, somma2=0;
21     for (int i=0; i<N ; i++) {
22         double x,y;
23         // Uso l'oggetto generatore di numeri random per generare numeri random in maniera uniforme
24         x=R.Uniform()*9+1;
25         y=f(x);
26         somma+=y;
27         somma2+=y*y;
28     }
29     // Calcolo medie
30     double media=somma/N;
31     double media2=somma2/N;
32     // Calcolo Area
33     double area=media*(10.-1.);
34     // Calcolo errore
35     double sigma=sqrt(media2-media*media);
36     double errore=sigma*(10.-1.)/sqrt(N);
37     printf("Integrale = %lf +/- %lf\n", area, errore);
38 }

```

Una formula alternativa per la varianza è

$$\sigma_X^2 = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

Questa formula è più pratica per calcolare la varianza.



---

GENERATORE DI NUMERI PSEUDO-CASUALI  
NON UNIFORMEMENTE DISTRIBUITI  
ANCORA SUL  
**METODO DI REIEZIONE**

---

# GENERATORI PER PDF NON UNIFORMI / RIGETTO

- Nel simulare un processo spesso è necessario estrarre numeri casuali in modo tale da seguire una specifica distribuzione.
- Serve, cioè, che i numeri estratti seguano una distribuzione di probabilità assegnata.
- Nei corsi precedenti avete incontrato alcune distribuzioni probabilità molto usate (Binomiale, Poisson, Gauss), ma in generale possono esserci processi aleatori che seguono funzioni di probabilità di tipo diverso.
  - Molti processi fisici sono processi stocastici
- Per simulare questi processi, quindi, occorre estrarre numeri casuali in modo tale che essi seguano un'arbitraria distribuzione di probabilità.
- Una qualunque funzione definita in un intervallo  $(a,b)$ , continua e derivabile  $N$  volte, definita positiva e tale che il suo integrale in  $(a,b)$  sia pari ad 1 può rappresentare una distribuzione di probabilità di una qualche variabile aleatoria (o processo fisico)

$$g(x) \geq 0 \quad \forall x \in [a, b]; \quad \int_a^b g(x) dx = 1$$

# GENERATORI PER PDF NON UNIFORMI / RIGETTO

- L'idea di base è molto semplice: ***dato un insieme di numeri casuali, ne vogliamo eliminare una parte in maniera che i rimanenti siano distribuiti secondo una certa densità di probabilità.***
- Supponiamo di voler estrarre numeri che seguano la distribuzione di probabilità  $g(x)$  definita in  $(a,b)$  e supponiamo che in  $(a,b)$  la funzione abbia massimo  $g_{\max}$  o che esista un  $g_{\max}$  tale che

$$g(x) < g_{\max} \quad \forall x \in [a, b]$$

- 1) estraiamo in maniera causale e secondo una distribuzione uniforme  $x_i$  nell'intervallo  $[a, b]$
- 2) definiamo un criterio di accettazione per  $x_i$  tale che gli  $x_i$  accettati risultino distribuito come la pdf che si vuole emulare:
  - Come ?
    - La probabilità di accettazione di  $x_i$  deve essere proporzionale a  $g(x_i)$

# GENERATORI PER PDF NON UNIFORMI / RIGETTO

- La **sequenza ottenuta**  $\{x_i'\}$  risulta distribuita secondo una **distribuzione di probabilità** pari al prodotto delle probabilità di estrazione e di accettazione

$$\frac{1}{b-a} dx$$

**Probabilità di estrazione di  $x_i$**

Estratto  $y_i$  tra 0 e  $g_{max}$  secondo una distribuzione uniforme,  
probabilità che  $y_i < g(x_i)$

$$g(x_i) / g_{max}$$

**Probabilità di accettazione di  $x_i$**

$$p(x) dx = \frac{1}{b-a} g(x_i) / g_{max} dx$$

$p(x)$  e  $g(x)$   
differiscono solo per  
la normalizzazione

# GENERATORI PER PDF NON UNIFORMI / RIGETTO

## Osservazione

- La **sequenza ottenuta**  $\{x_i'\}$  risulta distribuita secondo una **distribuzione di probabilità** pari al prodotto delle probabilità di estrazione e di accettazione

$\frac{1}{b-a} dx$       Probabilità di estrazione di  $x_i$   
 probabilità che  $y_i < g(x_i)$        $g(x_i)/g_{max}$       Probabilità di accettazione di  $x_i$

$$p(x)dx = \frac{1}{b-a} g(x_i)/g_{max} dx$$

NOTA: ritroviamo il metodo di integrazione con la tecnica del rigetto

$$I = \int_a^b g(x)dx = (b-a)g_{max} \int_a^b p(x)dx = A_{rettangolo} \frac{N_{accettati}}{N}$$

# GENERATORI PER PDF NON UNIFORMI / RIGETTO

## In pratica

- Si estraggono punti uniformemente distribuiti su un rettangolo che racchiude la funzione densità di probabilità  $g(x)$  che si vuole riprodurre
- Si accettano come valori random validi della variabile  $x$ , le ascisse dei punti che cadono sotto la curva della funzione densità di probabilità.

```

8 //
9 // Estraggo numeri secondo questa distribuzione di probabilità
10 double f(double x) {
11     return (2.*x*x+1.)/675.;
12 }
13 void Generatore(){
14     // creo un "oggetto" generatore di numeri random
15     TRandom R(345345);
16     int N=1000000;
17     // creo un oggetto file lo collego al file di nome Reiezione.root
18     // e gli dico di crearlo (ricrearlo) quindi è un output
19     TFile f("Reiezione.root","RECREATE");
20     TH1D * h1 = new TH1D ("h1","Istogramma di frequenze",80,0.,11.);
21     // inizializzo sequenza di conteggio
22     int In=0;
23     for (int i=0; i<N ; i++) {
24         double x,y;
25         // Uso l'oggetto generatore di numeri random per generare numeri random in maniera uniforme
26         x=R.Uniform()*9+1;
27         y=R.Uniform()*201;
28
29         // Riempio l'oggetto histogramma con solo i punti contenuti
30         if (y<f(x)) {
31             h1->Fill(x);
32             In++;
33         }
34     }
35     printf("Ho estratto n=%g valori utili\n",In);
36     // scrivo gli oggetti histogramma nel file
37     h1->Write();
38     // chiudo il file attraverso il metodo dell'oggetto di tipo file.
39     f.Close();
40 }

```



---

# GENERATORE DI NUMERI PSEUDO-CASUALI NON UNIFORMEMENTE DISTRIBUITI

-

## **METODO DELL'INVERSIONE**

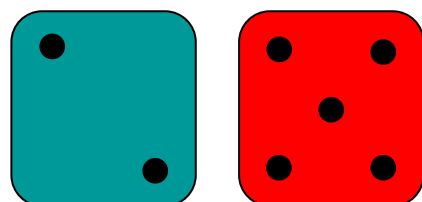
---

# GENERATORI PER PDF NON UNIFORMI / INVERSIONE

- Il metodo di reiniezione è semplice ma scarsamente efficiente.
- Se si vogliono ottenere  $N$  valori estratti casualmente occorre generare molto più di  $2N$  numeri casuali. Esistono altri algoritmi che permettono, per certi tipi di distribuzioni, di ottenere gli  $N$  valori ricorrendo a un numero molto inferiore di estrazioni.
- Cominciamo col considerare una variabile aleatoria discreta, cioè in grado di assumere solo un numero  $N$  finiti di valori.

Quale è la distribuzione di probabilità della somma dei numeri sulle facce di due dadi?

$x_i$	2	3	4	5	6	7	8	9	10	11	12
$P(x_i)$	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36



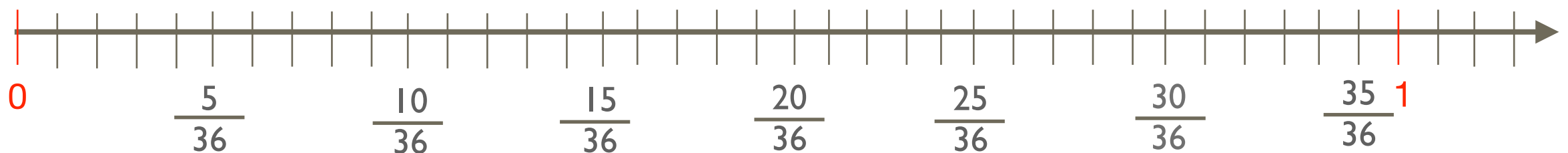
# GENERATORI PER PDF NON UNIFORMI / INVERSIONE

- Se costruisco una successione di segmenti ognuno di lunghezza  $p(x_i)$  otterrò un segmento la cui lunghezza totale sarà 1

- Infatti 
$$\sum_{i=1}^{12} p(x_i) = 1$$

$x_i$	2	3	4	5	6	7	8	9	10	11	12
$p(x_i)$	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36

2 3 4 5 6 7 8 9 10 11 12



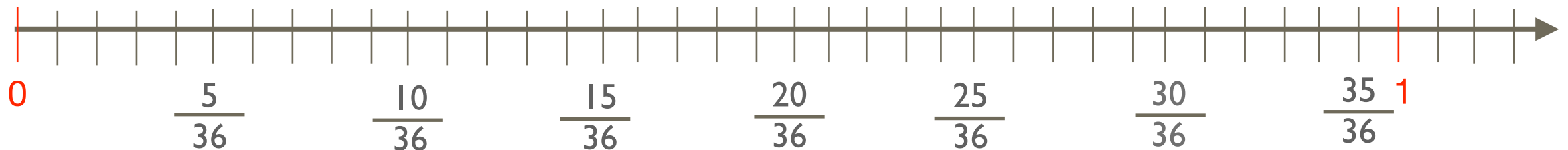
Lunghezza totale = 1

# GENERATORI PER PDF NON UNIFORMI / INVERSIONE

- Se estraggo un numero casuale  $\eta_i$  uniformemente tra 0 e 1 questo cadrà in un sotto-segmento con probabilità proporzionale alla lunghezza del sotto-segmento.
- Quindi, **gli esiti  $x_i$** , estratti sulla base dell'intervallo in cui cade  $\eta_i$  (uniformemente distribuito in  $[0,1]$ ) **seguono la distribuzione di probabilità discreta  $p(x_i)$** .

$x_i$	2	3	4	5	6	7	8	9	10	11	12
$p(x_i)$	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36

2 3 4 5 6 7 8 9 10 11 12



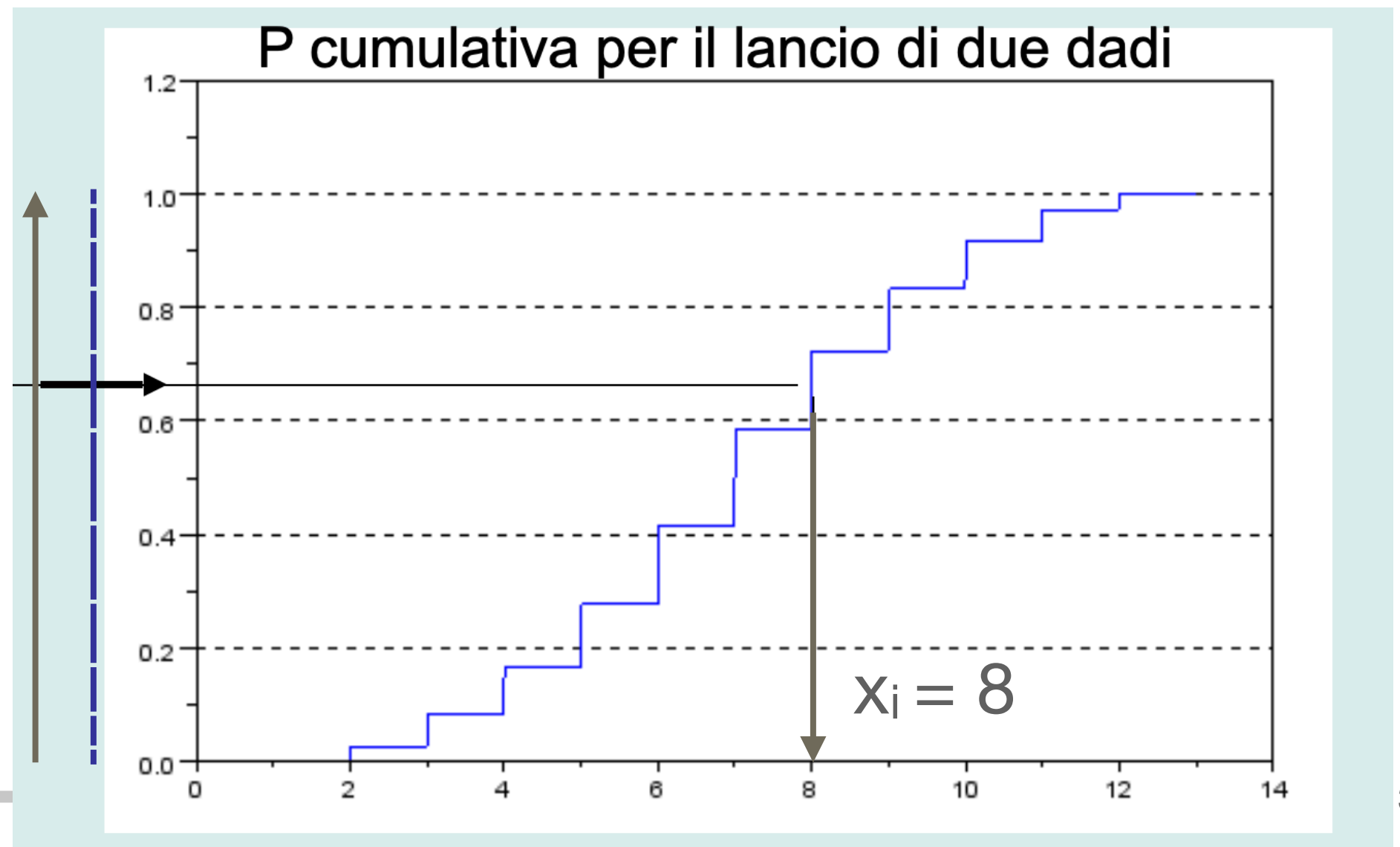
Lunghezza totale = 1

# GENERATORI PER PDF NON UNIFORMI / INVERSIONE

- Consideriamo ora la **probabilità cumulativa**, così definita:
  - la probabilità cumulativa del valore  $x_k$  è la probabilità di ottenere un qualsiasi valore compreso tra  $x_1$  a  $x_k$
  - questa è data da:  $P_k = \sum_{i=1}^k p(x_i)$
  - ovviamente per  $k=N$  avremmo  $P(x_N) = 1$

$$\eta_i = 0.66$$

Lunghezza totale = 1





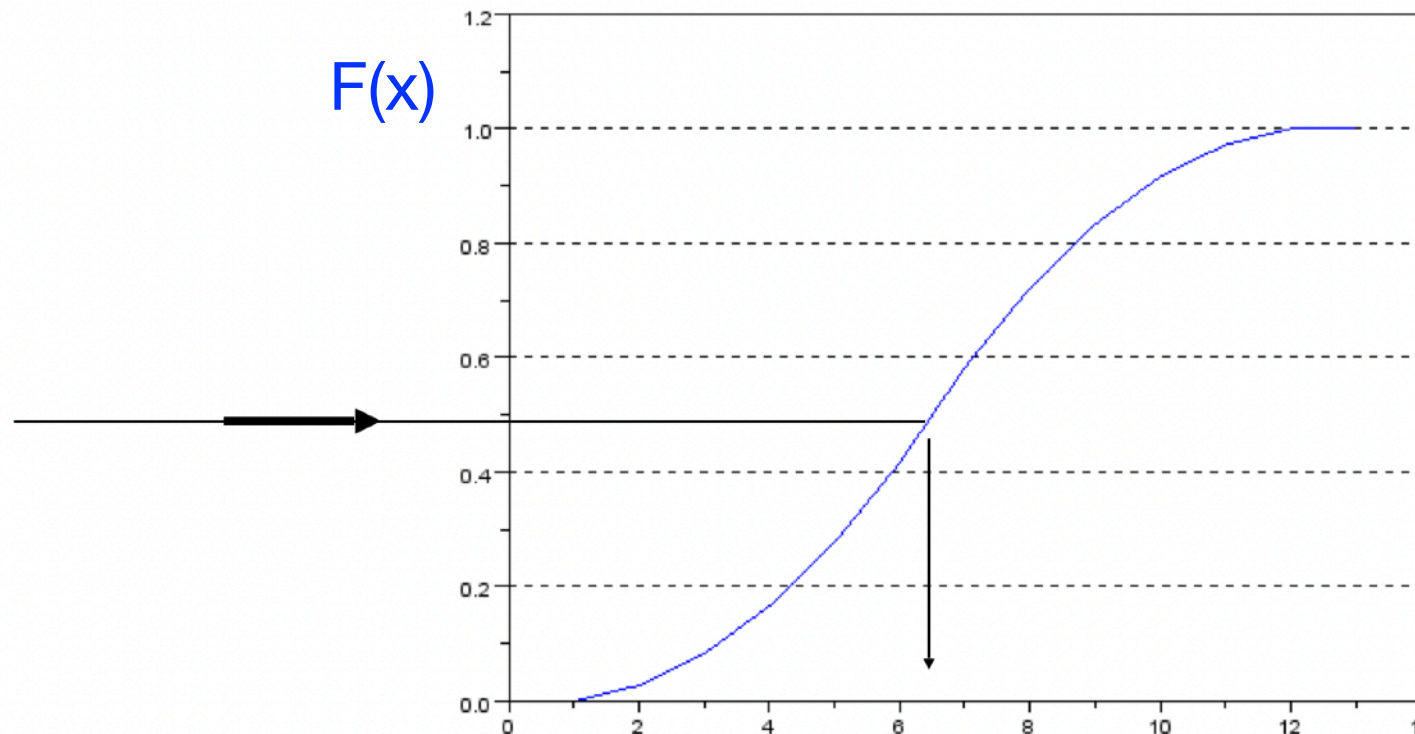
# GENERATORI PER PDF NON UNIFORMI / INVERSIONE

Nel caso di variabili aleatorie continue posso utilizzare lo stesso metodo. Supponiamo di avere una distribuzione di probabilità  $g(x)$  di una variabile aleatoria continua. La sua distribuzione cumulativa sarà:

Cumulativa di  $g(x)$  
$$F(x) = \int_{-\infty}^x g(t) dt$$

voglio generare  
secondo la pdf  $g(x)$

Trattandosi di una distribuzione cumulativa essa sarà una funzione monotona crescente che assume valori in  $[0, 1]$ .



Posso  
procedere  
allo stesso  
modo.



# GENERATORI PER PDF NON UNIFORMI / INVERSIONE

- Come fare passo per passo:

- ***Fase preliminare analitica:***

- Determinare una primitiva (=cumulativa F) della distribuzione di probabilità che si vuole campionare

$$y = F(x) = \int_{-\infty}^x g(t)dt$$

- Invertire la primitiva:

$$h(y) = F^{-1}(y)$$

- ***Fase numerica di estrazione dei valori***

- Estrarre un numero casuale  $y$  in maniera uniforme in  $(0,1)$
- Calcolare  $x = h(y)$  a partire da  $y$  utilizzando l'inversa della primitiva

# GENERATORI PER PDF NON UNIFORMI / INVERSIONE

## ESEMPIO

Consideriamo la distribuzione di probabilità:

$$g(x) = 2x \quad [0,1]$$

Definita nell'intervallo  $[0,1]$ .

Questa possiede tutti i requisiti di una distribuzione di probabilità.

- Definita positiva
- Integrale uguale a 1
- Continua e derivabile.

Calcolarne una sua primitiva è semplice così come è semplice invertire quest'ultima.

Primitiva  $y=F(x)= x^2$

$$x = F^{-1}(y) = \sqrt{y}$$

Inversa della primitiva

# GENERATORI PER PDF NON UNIFORMI / INVERSIONE

- Si determina analiticamente l'inversa della funzione cumulativa.
- Si estraggono numeri casuali in maniera uniforme tra 0 e 1
- Si utilizza la funzione inversa per ottenere il valore casuale voluto.

```

8 //
9 // estraggo valori secondo una distribuzione data
10 // usando il metodo dell'inversione
11 //
12 // distribuzione di probabilità o di frequenze
13 //
14 double f(double x) {
15     return 2.*x;
16 }
17 double inv(double y) {
18     return sqrt(y);
19 }
20 void Inversione() {
21     // creo un "oggetto" generatore di numeri random
22     TRandom R(12345);
23     int N=100000;
24     // creo un oggetto file lo collego al file di nome inversione.root
25     // e gli dico di crearlo (ricrearlo) quindi è un output
26     TFile f("inversione.root", "RECREATE");
27     // creo un oggetto histogramma
28     TH1D * h = new TH1D ("h", "Istogramma delle Frequenze", 80, 0, 1.);
29     for (int i=0; i<N ; i++) {
30         double x,y;
31         //estraggo numeri in maniera uniforme tra 0 e 1
32         y=R.Uniform();
33         // uso l'inversa della primitiva
34         x=inv(y);
35
36         // Riempio l'oggetto histogramma con i numeri random generati
37         h->Fill(x);
38     }
39     // scrivo l'oggetto histogramma nel file
40     h->Write();
41     // chiudo il file attraverso il metodo dell'oggetto di tipo file.
42     f.Close();
43 }

```

Macro:  
GenRandom\_MetodoInv.C

---

**FINE**

---