

# Υπολογιστικές Μέθοδοι

<http://eclass.uoa.gr/courses/PHYS186/>

Διδάσκοντες:

Φ. Διάκονος

Δ. Φασουλιώτης

➤ Μέθοδοι Monte-Carlo και εφαρμογές

# Μέθοδοι Monte Carlo

## **Τι είναι:**

Οποιαδήποτε αριθμητική μέθοδος  
χρησιμοποιεί ψευδοτυχαίους αριθμούς

## **Που χρησιμοποιείται:**

### **Παντού**

Ολοκλήρωση

Επίλυση διαφορικών εξισώσεων

Προσομοίωση στοχαστικών συστημάτων

## **Γιατί;**

Εύκολη στην εφαρμογή

Εφαρμόσιμη και με τις πιο περίπλοκες συνοριακές συνθήκες

Αποδοτική σε πολυδιάστατα προβλήματα

# Μέθοδοι MC

## Πως υλοποιούνται:

- Επιλογή της συνάρτησης πυκνότητας πιθανότητας (pdf) του υποθετικού πληθυσμού.
- Γέννηση τυχαίου δείγματος από μια ακολουθία ψευδοτυχαίων αριθμών σύμφωνα με την παραπάνω πιθανότητα.
- Στατιστική επεξεργασία για την εκτίμηση των υπό εξέταση παραμέτρων από το τυχαίο δείγμα.

# Μέθοδοι MC: Τυχαίοι αριθμοί

- **Τυχαίοι αριθμοί**

Φυσικές στοχαστικές διαδικασίες

(ηλεκτρονικός θόρυβος, πυρηνικές διασπάσεις, κ.λ.π.)

- **Ψευδοτυχαίοι αριθμοί**

Παράγονται αιτιοκρατικά από υπολογιστικούς αλγορίθμους

Εύκολοι στην παραγωγή

Επαναλήψιμη διαδικασία

Παρέχουν ομοιόμορφη κάλυψη φασικού χώρου

- **Σχεδόν Ψευδοτυχαίοι αριθμοί**

Παράγονται αιτιοκρατικά από υπολογιστικούς αλγορίθμους

Αλυσίδες Markov

Παρέχουν προμελετημένη κάλυψη φασικού χώρου

# Μέθοδοι MC: Τυχαίοι αριθμοί

**Αλγόριθμοι (Γεννήτορες) Ψευδοτυχαίων Αριθμών**  
Ομοιόμορφη κατανομή [0.,1.]

**Χαρακτηριστικά:**

- Περίοδος
- Συσχέτιση
- Απόδοση

**Υλοποίηση:**

$$I_{j+1} = \text{mod} ( aI_j + c, m )$$

... και ποικίλες παραλλαγές

**Ποιοτικός έλεγχος:**

- Θεωρητικός έλεγχος
- Πρακτικός έλεγχος

Περισσότερες πληροφορίες  
<http://random.mat.sbg.ac.at>  
**Numerical Recipes**  
κ.λ.π

*Παράδειγμα γεννήτορα ψευδο-τυχαίων αριθμών*

$$x_{n+1} = (a x_n + c) \bmod m$$

με  $a = 1103515245$ ,  $c = 12345$  and  $m = 2^{31}$ .

```
int x=1717; //seed – αρχική τιμή της ακολουθίας ψευδο-τυχαίων αριθμών  
           // μπορεί να επιλέγεται με διάφορους τρόπους  
int m=pow(2,31);  
x=(1103515245*x+12345)%m;  
double random=double(x)/m;
```

## Numerical Recipes

```
#define IA 16807
#define IM 2147483647
#define AM (1.0/IM)
#define IQ 127773
#define IR 2836
#define MASK 123459876
```

```
float ran0(long *idum)
```

“Minimal” random number generator of Park and Miller. Returns a uniform random deviate between 0.0 and 1.0. Set or reset `idum` to any integer value (except the unlikely value `MASK`) to initialize the sequence; `idum` must not be altered between calls for successive deviates in a sequence.

```
{
    long k;
    float ans;

    *idum ^= MASK;
    k=(*idum)/IQ;
    *idum=IA*(*idum-k*IQ)-IR*k;
    if (*idum < 0) *idum += IM;
    ans=AM*(*idum);
    *idum ^= MASK;
    return ans;
}
```

XORing with `MASK` allows use of zero and other simple bit patterns for `idum`.

Compute `idum=(IA*idum) % IM` without overflows by Schrage’s method.

Convert `idum` to a floating result.

Unmask before return.

## Numerical Recipes

```
#define IA 16807
#define IM 2147483647
#define AM (1.0/IM)
#define IQ 127773
#define IR 2836
#define NTAB 32
#define NDIV (1+(IM-1)/NTAB)
#define EPS 1.2e-7
#define RNMIX (1.0-EPS)
```

```
float ran1(long *idum)
```

"Minimal" random number generator of Park and Miller with Bays-Durham shuffle and added safeguards. Returns a uniform random deviate between 0.0 and 1.0 (exclusive of the endpoint values). Call with `idum` a negative integer to initialize; thereafter, do not alter `idum` between successive deviates in a sequence. `RNMIX` should approximate the largest floating value that is less than 1.

```
{
    int j;
    long k;
    static long iy=0;
    static long iv[NTAB];
    float temp;

    if (*idum <= 0 || !iy) {
        if (-(*idum) < 1) *idum=1;
        else *idum = -(*idum);
        for (j=NTAB+7; j>=0; j--) {
            k=(*idum)/IQ;
            *idum=IA*(*idum-k*IQ)-IR*k;
            if (*idum < 0) *idum += IM;
            if (j < NTAB) iv[j] = *idum;
        }
        iy=iv[0];
    }
    k=(*idum)/IQ;
    *idum=IA*(*idum-k*IQ)-IR*k;
    if (*idum < 0) *idum += IM;
    j=iy/NDIV;
    iy=iv[j];
    iv[j] = *idum;
    if ((temp=AM*iy) > RNMIX) return RNMIX;
    else return temp;
}
```

Initialize.

Be sure to prevent `idum = 0`.

Load the shuffle table (after 8 warm-ups).

Start here when not initializing.

Compute `idum=(IA*idum) % IM` without overflows by Schrage's method.

Will be in the range `0..NTAB-1`.

Output previously stored value and refill the shuffle table.

Because users don't expect endpoint values.



# Τυχαίοι αριθμοί: Έλεγχος

Γεννήτορες Ψευδοτυχαίων Αριθμών: Πρακτικός έλεγχος

Υπολογισμός των ροπών  $k$  τάξης  
της κατανομής τυχαίων αριθμών:  $\langle x^k \rangle = \frac{1}{n} \sum_{i=1}^n x_i^k$

Αν οι τυχαίοι αριθμοί κατανέμονται με ομοιόμορφη πιθανότητα  $p(x)$

$$\frac{1}{n} \sum_{i=1}^n x_i^k \approx \int_0^1 dx x^k p(x) + \mathcal{O}(1/\sqrt{n}) \approx \frac{1}{k+1} + \mathcal{O}(1/\sqrt{n})$$

# Τυχαίοι αριθμοί: Έλεγχος

Γεννήτορες Ψευδοτυχαίων Αριθμών: Πρακτικός έλεγχος

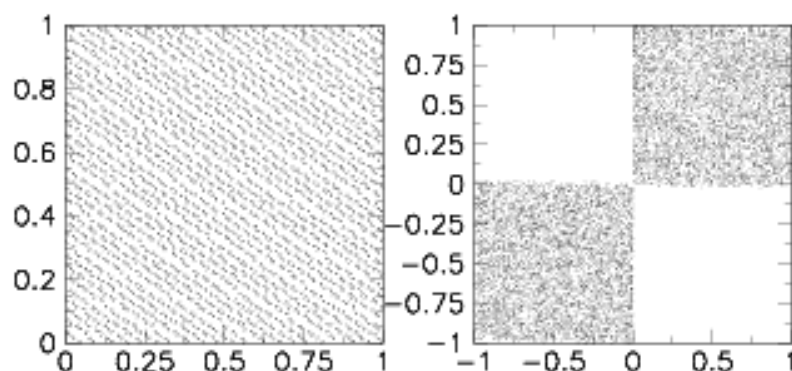
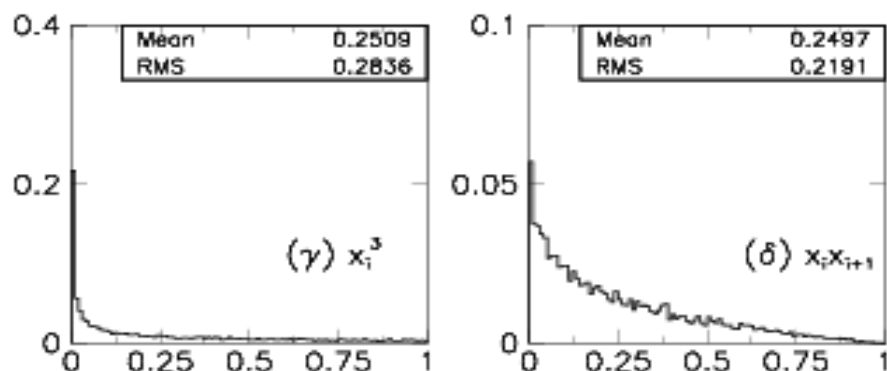
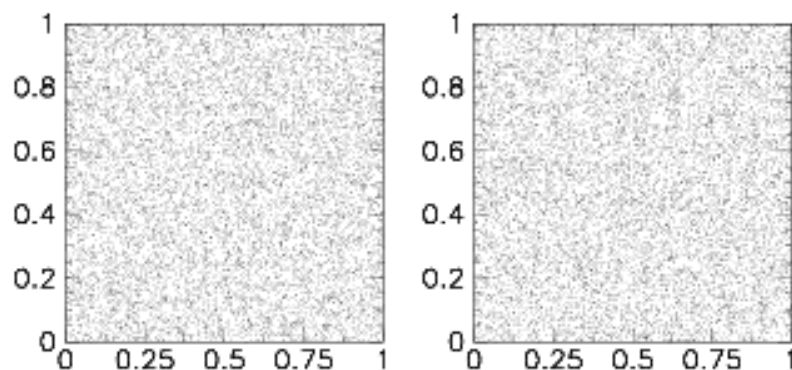
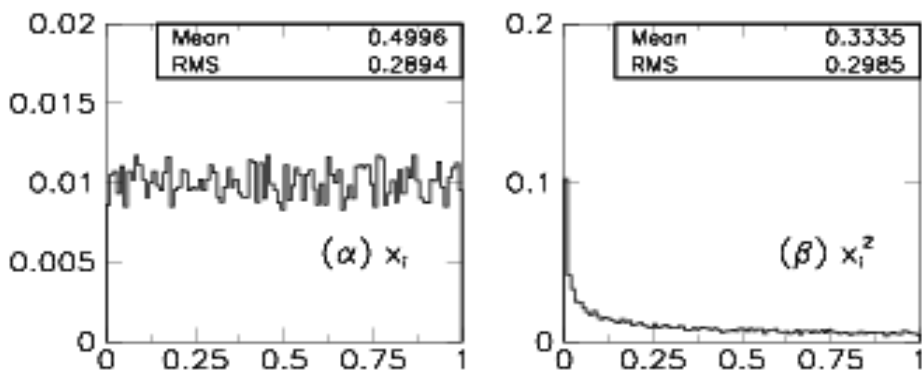
Υπολογισμός των Συσχετίσεων: 
$$C(k) = \frac{1}{n} \sum_{i=1}^n x_i x_{i+k}$$

Αν οι τυχαίοι αριθμοί είναι ανεξάρτητοι  
και κατανέμονται ομοιόμορφα

$$\frac{1}{n} \sum_{i=1}^n x_i x_{i+k} \approx \int_0^1 dx \int_0^1 dy x y p(x, y) = \frac{1}{4}$$

# Τυχαίοι αριθμοί: Έλεγχος

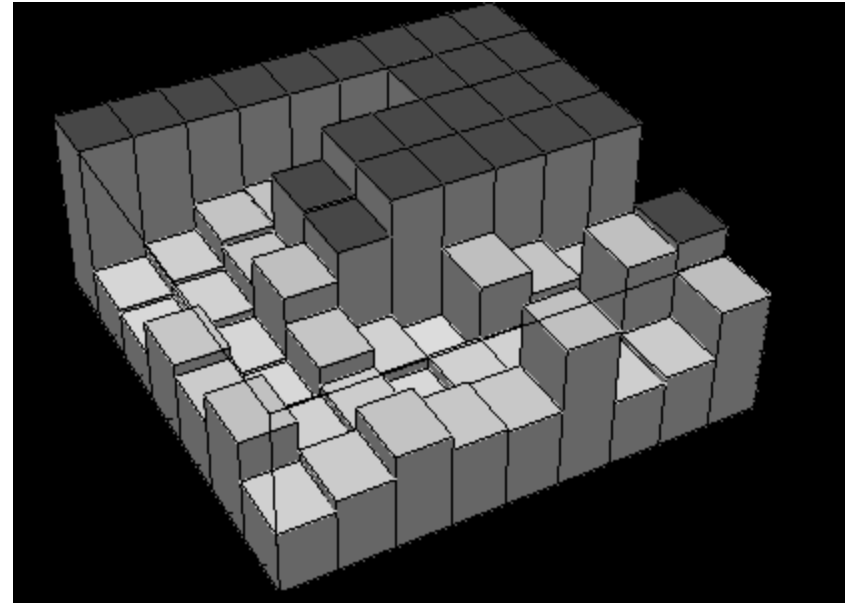
## Γεννήτορες Ψευδοτυχαίων Αριθμών: Πρακτικός έλεγχος



# Τυχαίοι αριθμοί: Έλεγχος

Γεννήτορες Ψευδοτυχαίων Αριθμών: Πρακτικός έλεγχος

Πιο σύνθετοι έλεγχοι:  
π.χ. Κατανομή  $\kappa$  αριθμών  
σε επίπεδα  $\kappa-1$  διαστάσεων



Τελικός έλεγχος:

Επανάληψη της μελέτης με άλλο γεννήτορα τυχαίων αριθμών

Γεννήτορες Ψευδοτυχαίων Αριθμών μη ομοιόμορφης κατανομής  
Μέθοδος μετασχηματισμού

Έστω  $x$  τυχαία μεταβλητή ομοιόμορφα κατανεμημένη στο διάστημα  $[0, 1]$

$$p(x) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & \text{αλλού} \end{cases}$$

Αν μετασχηματίσουμε τη μεταβλητή  $x \rightarrow y$ , η πιθανότητα πρέπει να διατηρείται

Οπότε  $p(y)dy = p(x)dx$

και δεδομένου ότι  $p(x)=1$ , τότε

$$p(y)dy = dx \Rightarrow x(y) = \int_0^y p(y')dy'$$

## Παράδειγμα 1

Έστω ότι θέλουμε  $y$  να κατανέμεται ομοιόμορφα στο διάστημα  $[a, b]$

$$p(y) = \frac{1}{b-a}, \quad a \leq y \leq b$$

τότε

$$p(y)dy = \frac{dy}{b-a} = dx \Rightarrow$$

$$x(y) = \int_a^y \frac{dy'}{b-a} \Rightarrow x = \frac{1}{b-a} (y-a) \Rightarrow$$

$$y = a + (b-a)x$$

## Παράδειγμα 2

Έστω ότι θέλουμε  $y$  να κατανέμεται γραμμικά

$$p(y) \propto y$$

$$p(y) = 2y, \quad 0 \leq y \leq 1$$

τότε

$$p(y)dy = 2ydy = dx \Rightarrow$$

$$x(y) = \int_0^y 2y'dy' \Rightarrow x = y^2 \Rightarrow$$

$$y = \sqrt{x}$$

### Παράδειγμα 3

Έστω ότι θέλουμε  $y$  να κατανέμεται εκθετικά

$$p(y)dy = e^{-y}$$

τότε

$$p(y)dy = e^{-y}dy = dx \Rightarrow$$

$$x(y) = \int_0^y e^{-y'} dy' \Rightarrow x = 1 - e^{-y} \Rightarrow$$

$$y = -\ln(1 - x)$$



## Μέθοδος μετασχηματισμού – Κανονική κατανομή

Οι μέθοδοι μετασχηματισμού γενικεύονται και σε περισσότερες διαστάσεις

$$p(y_1, y_2, \dots) dy_1 dy_2 \dots = p(x_1, x_2, \dots) \left| \frac{\partial(x_1, x_2, \dots)}{\partial(y_1, y_2, \dots)} \right| dy_1 dy_2 \dots$$

Για παράδειγμα αν

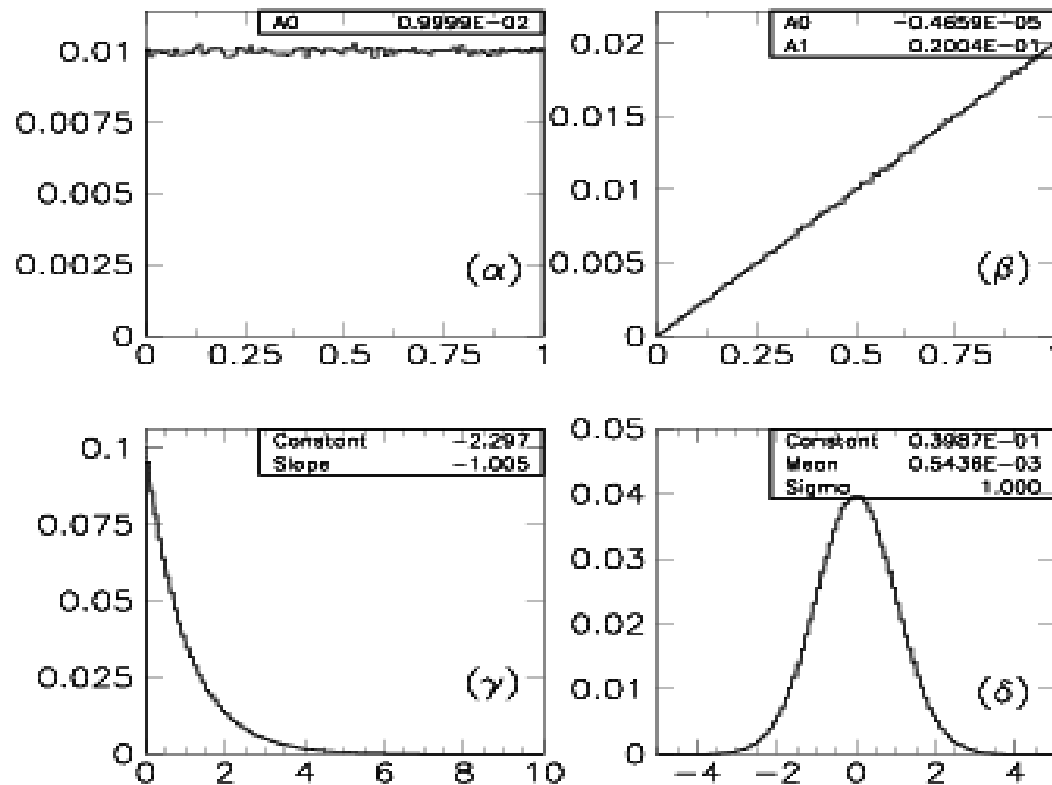
$$y_1 = \sqrt{-2 \ln x_1} \cos 2\pi x_2 \quad y_2 = \sqrt{-2 \ln x_1} \sin 2\pi x_2$$

τότε

$$\left| \frac{\partial(x_1, x_2)}{\partial(y_1, y_2)} \right| = - \left( \frac{1}{\sqrt{2\pi}} e^{-y_1^2/2} \right) \left( \frac{1}{\sqrt{2\pi}} e^{-y_2^2/2} \right)$$

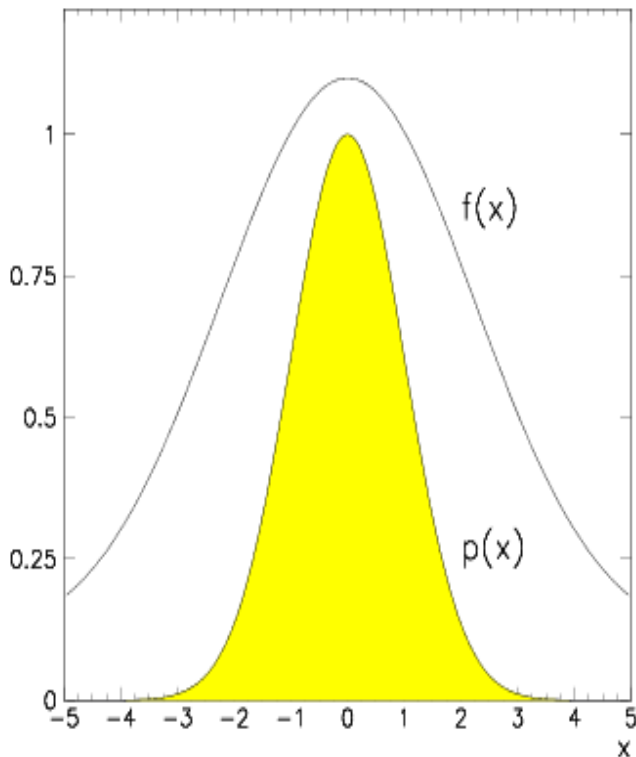
# Τυχαίοι αριθμοί: Μεθ. Μετασχηματισμού

Γεννήτορες Ψευδοτυχαίων Αριθμών μη ομοιόμορφης κατανομής  
Μέθοδος μετασχηματισμού



# Τυχαίοι αριθμοί: Μεθ. Απόρριψης

## Γεννήτορες Ψευδοτυχαίων Αριθμών μη ομοιόμορφης κατανομής Μέθοδος απόρριψης



- Γεννάμε σημεία σε δύο διαστάσεις τα οποία είναι ομοιόμορφα κάτω από τη συνάρτηση σύγκρισης.
- Όποτε το σημείο βρίσκεται κάτω από τη συνάρτηση πιθανότητας το κρατάμε, αν όχι το απορρίπτουμε και γεννάμε καινούριο σημείο.
- Με αυτή τη διαδικασία έχουμε καταφέρει να γεννάμε ομοιόμορφα σημεία κάτω από την  $p(x)$ .

# Τυχαίοι αριθμοί: Μεθ. Απόρριψης

## Γεννήτορες Ψευδοτυχαίων Αριθμών μη ομοιόμορφης κατανομής Μέθοδος απόρριψης

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <TFile.h>
#include <TH1F.h>
#include <TRandom.h>

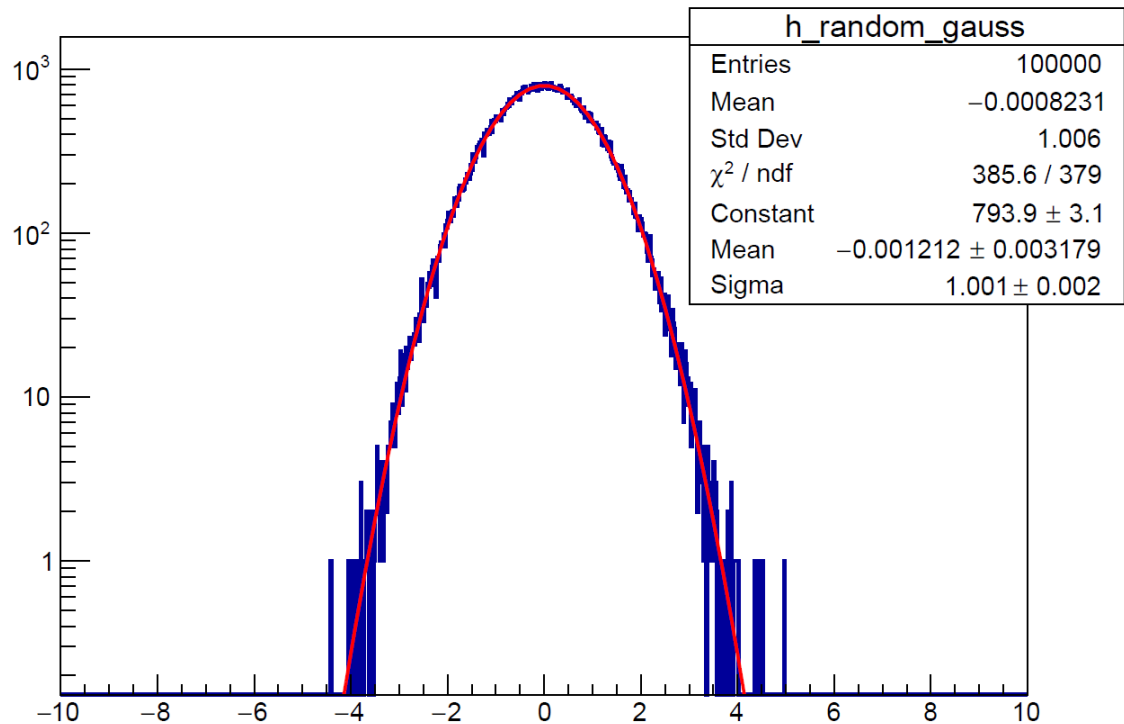
void random_gauss(int N, int seed)
{
    TH1F *h=new TH1F("h_random_gauss"," ",1000,-10.,10.);
    TRandom *ran=new TRandom3(seed);
    int i=0;
    while(i<N)
    {
        double x=10.*ran->Rndm()-5.;
        double y=ran->Rndm();
        double rg=exp(-0.5*x*x);
        if(rg>y){ i++; h->Fill(x);}
    }
    h->SetLineWidth(2.);
    //h->SetLineColor(kRed);
    h->Draw("");
    h->Fit("gaus");
}
```

# Τυχαίοι αριθμοί: Μεθ. Απόρριψης

## Γεννήτορες Ψευδοτυχαίων Αριθμών μη ομοιόμορφης κατανομής Μέθοδος απόρριψης

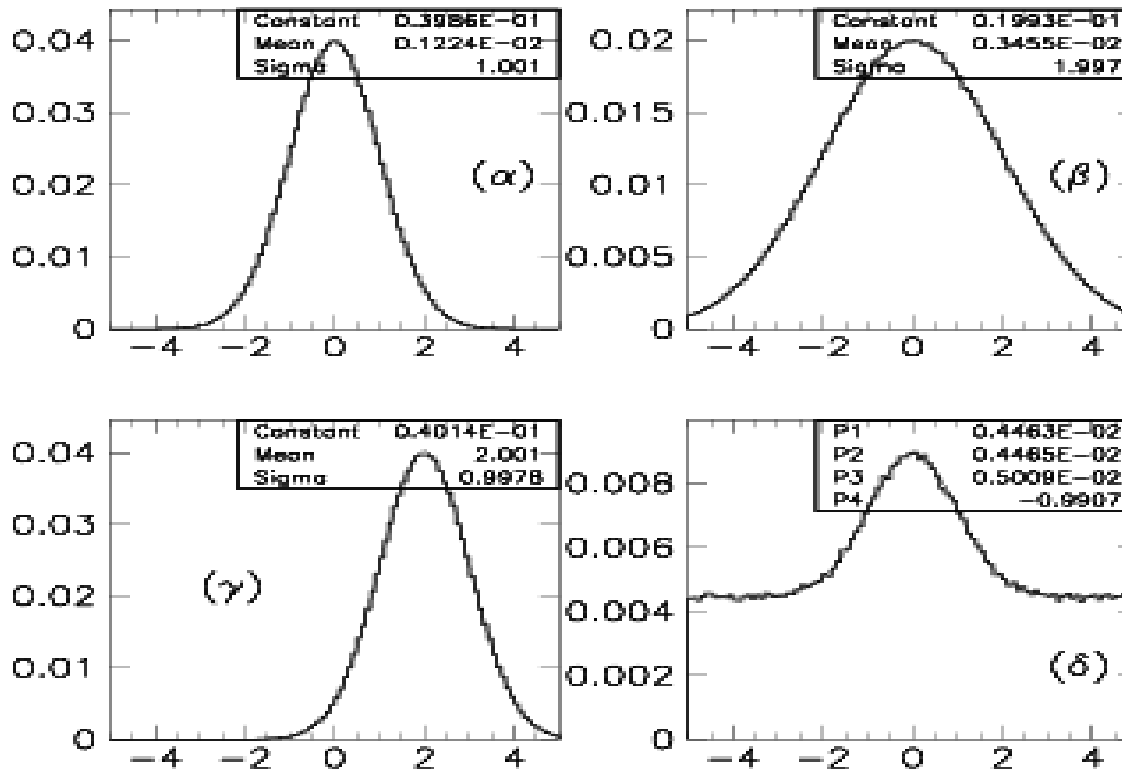
```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <TFile.h>
#include <TH1F.h>
#include <TRandom.h>

void random_gauss(int N, int seed)
{
    TH1F *h=new TH1F("h_random_gauss");
    TRandom *ran=new TRandom3(seed);
    int i=0;
    while(i<N)
    {
        double x=10.*ran->Rndm()-5.;
        double y=ran->Rndm();
        double rg=exp(-0.5*x*x);
        if(rg>y){ i++; h->Fill(x); }
    }
    h->SetLineWidth(2.);
    //h->SetLineColor(kRed);
    h->Draw("");
    h->Fit("gaus");
}
```



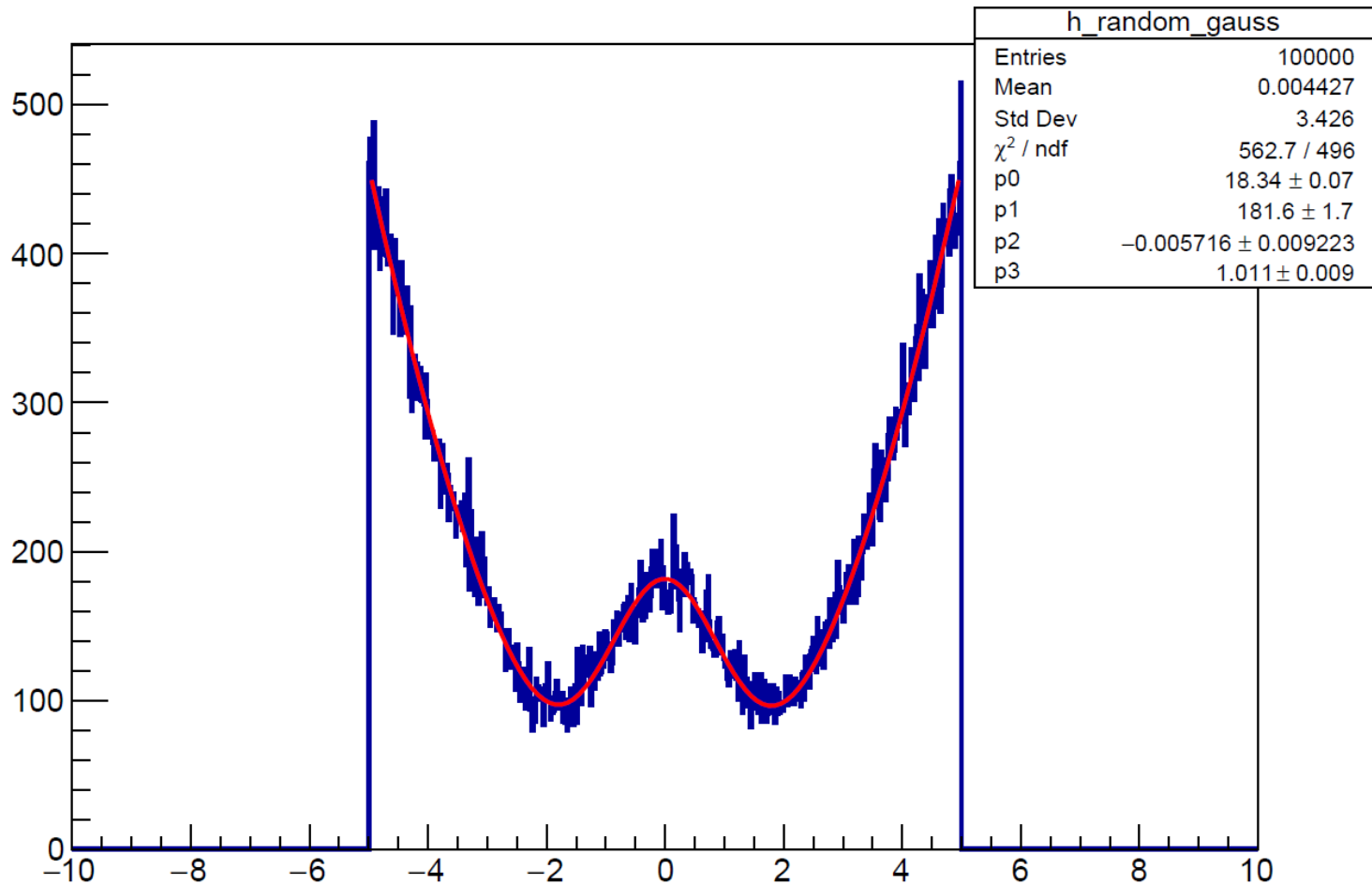
# Τυχαίοι αριθμοί: Μεθ. Απόρριψης

Γεννήτορες Ψευδοτυχαίων Αριθμών μη ομοιόμορφης κατανομής  
Μέθοδος απόρριψης



# Τυχαίοι αριθμοί: Μεθ. Απόρριψης

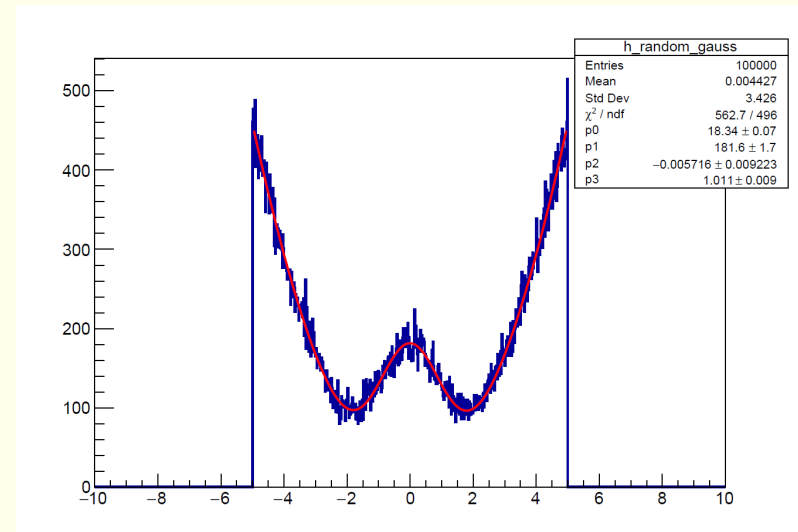
Γεννήτορες Ψευδοτυχαίων Αριθμών μη ομοιόμορφης κατανομής  
Μέθοδος απόρριψης



# Τυχαίοι αριθμοί: Μεθ. Απόρριψης

## Γεννήτορες Ψευδοτυχαίων Αριθμών μη ομοιόμορφης κατανομής Μέθοδος απόρριψης

```
double fitf(double *x,double *p) {  
    double val=p[0]*x[0]*x[0]+p[1]*exp(-0.5*(x[0]-p[2])*(x[0]-p[2])/p[3]/p[3]);  
    return val;  
}  
  
void random_gauss(int N, int seed)  
{  
    TH1F *h=new TH1F("h_random_gauss"," ",1000, -10., 10.);  
    TRandom *ran=new TRandom3(seed);  
    int i=0;  
    while(i<N)  
    {  
        double x=10.*ran->Rndm()-5.;  
        double y=2.5*ran->Rndm();  
        double rg=0.1*x*x+exp(-0.5*x*x);  
        if(rg>y){  
            i++; h->Fill(x);  
        }  
    }  
    h->SetLineWidth(2.);  
    h->Draw("");  
    TF1 *func=new TF1("func",fitf, -5., 5., 4);  
    func->SetParameters(100.,100.,0.,1.);  
    h->Fit("func"," ","",-5.,5.);  
}
```





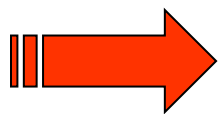
# MC Ολοκλήρωση: Crude

## Απλοϊκό (crude) Monte Carlo

$$I = \int_a^b y(x) dx \quad \Rightarrow \quad I = \frac{b-a}{n} \sum_{i=1}^n y(x_i)$$

Όπου τα  $x_i$  δεν είναι ισαπέχοντα πλέον, αλλά τυχαία κατανεμημένα

$$\mu_y = E[y] = \frac{1}{b-a} \int_a^b y(x) dx = \frac{I}{b-a} \quad \sigma_y^2 = V[y] = \frac{1}{b-a} \int_a^b y^2 dx - \mu_y^2$$



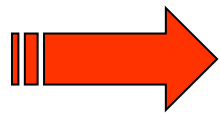
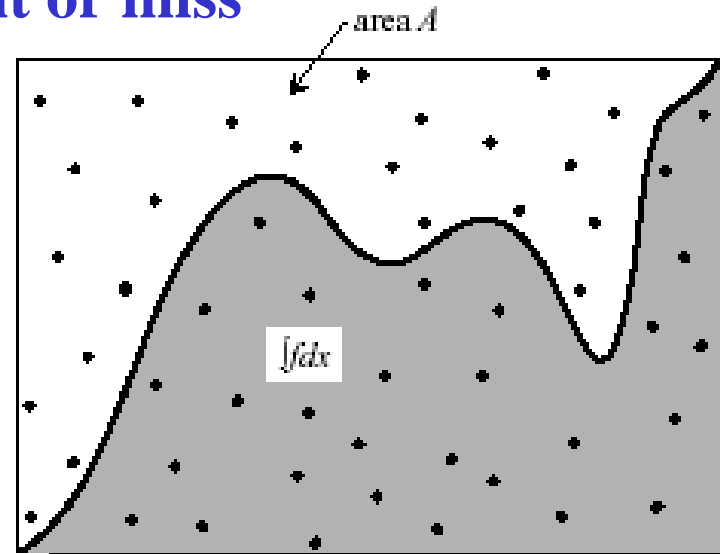
$$I = \frac{b-a}{n} \sum_{i=1}^n y(x_i) \pm (b-a) \frac{\sigma_y}{\sqrt{n}}$$

# MC Ολοκλήρωση: Hit or miss

## Monte Carlo hit or miss

Γεννάμε δύο ομοιόμορφα τυχαίους αριθμούς. Έναν  $x_i$  στο διάστημα  $[a,b]$  και έναν  $y_i$  στο διάστημα  $[y_{\min}, y_{\max}]$ . Το γεννημένο σημείο θεωρείται:

- επιτυχές (hit) αν  $y_i < y(x_i)$
- άστοχο (miss) αν  $y_i > y(x_i)$



$$I = \frac{n_{hit}}{n} (b - a) (y_{\max} - y_{\min}) + y_{\min} (b - a)$$

$$V[I] = \frac{1}{n^2} V[n_{hits}] (b - a)^2 (y_{\max} - y_{\min})^2$$

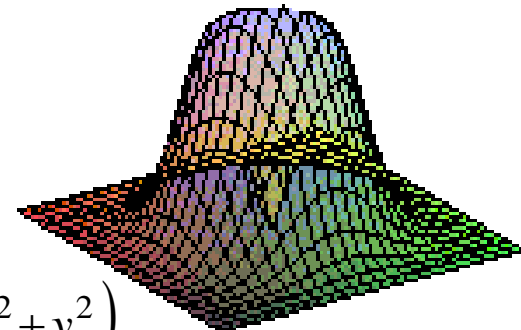
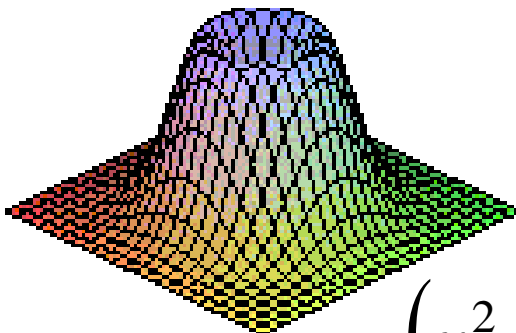
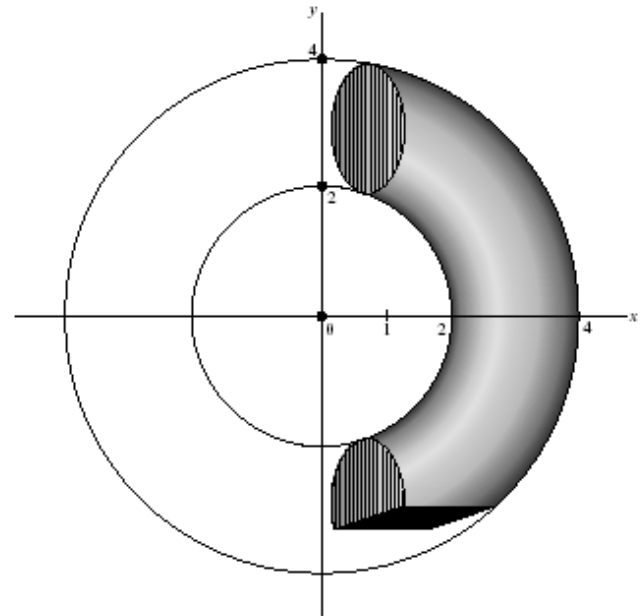
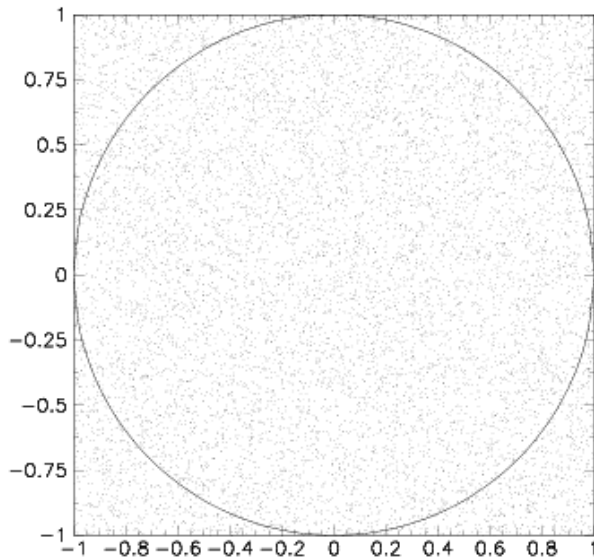
$$\text{με } V[n_{hits}] = np(1-p)$$



$$\frac{\Delta I}{I} = \frac{1}{\sqrt{n_{hit}}} \sqrt{\left(1 - \frac{n_{hit}}{n}\right)}$$

# MC Ολοκλήρωση: Παραδείγματα

## Monte Carlo hit or miss



$$(x^2 + y^2)e^{-(x^2 + y^2)}$$

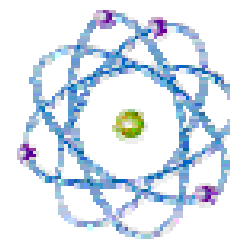
# MC Ολοκλήρωση: Σύγκριση

Μέθοδος	Αβεβαιότητα σαν συνάρτηση του αριθμού των σημείων	
	1 διάσταση	D διαστάσεις
Monte Carlo	$n^{-1/2}$	$n^{-1/2}$
Τραπεζίου	$n^{-2}$	$n^{-2/d}$
Simpson	$n^{-4}$	$n^{-4/d}$
Gauss τάξης m	$n^{-2m+1}$	$n^{-(2m-1)/d}$

# MC Ολοκλήρωση: Παραδείγματα

## Multidimensional Integration

Example: Atomic Physics



**<sup>4</sup>Be**

$$I = \int_0^1 dx_1 \int_0^1 dx_2 \dots \int_0^1 dx_{12} f(x_1, x_2, \dots, x_{12})$$

3 Dimension/electron \* 4 electrons = 12 Dimensions

For 100 points in each integration there are  $100^{12} = 10^{24}$  calculations

Assuming 1 Giga evaluations/sec

It would take over  $10^7$  years!!!!

# MC Ολοκλήρωση: Παραδείγματα

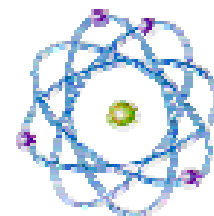
## Multidimensional Integration via MC mean value

Example: Atomic Physics

$$I = \int_0^1 dx_1 \int_0^1 dx_2 \dots \int_0^1 dx_{12} f(x_1, x_2, \dots, x_{12})$$

3 Dimension/electron \* 4 electrons = 12 Dimensions

$$\simeq (1-0)^{12} * \frac{1}{N} \sum_l^N f(x_1^l, x_2^l, \dots, x_{12}^l)$$



**<sup>4</sup>Be**

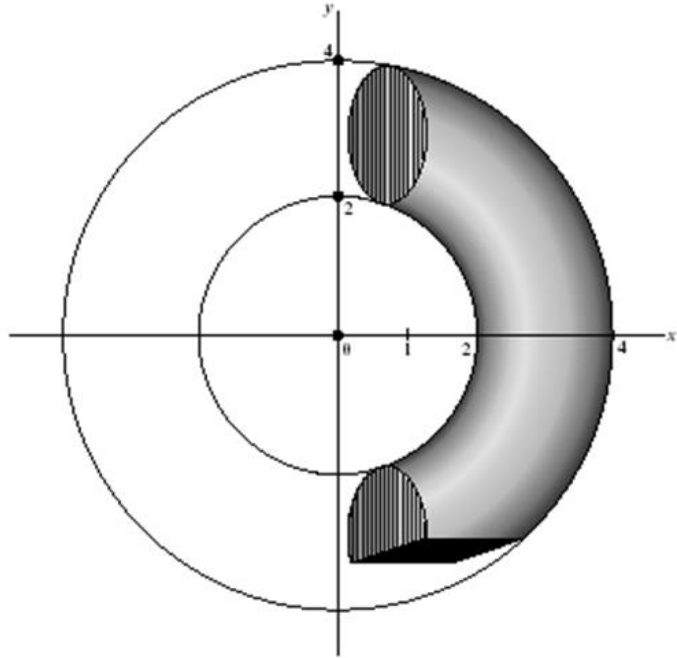
For  $N=10^6$  random points in the MC integration there are  $\sim 10^6$  calculations

Assuming 1 Giga evaluations/sec

It would take  $\sim 10^{-3}$  sec

**← compare to 7yrs**

## Μέθοδος απόρριψης - Ολοκλήρωμα με περίεργο σύνορο



$$\int \rho dx dy dz$$

$$\int x \rho dx dy dz$$

$$\int y \rho dx dy dz$$

$$\int z \rho dx dy dz$$

Υπολογισμός μάζας και κέντρου μάζας  
«περίεργου αντικειμένου»  
Έστω την τομή τοροειδούς με ορθογώνιο  
κουτί

$$z^2 + \left( \sqrt{x^2 + y^2} - 3 \right)^2 \leq 1$$

$$x \geq 1$$

$$y \geq -3$$

Όρια στα οποία κυμαίνονται τα  $x, y, z$

$$4 \geq x \geq 1$$

$$4 \geq y \geq -3$$

$$1 \geq z \geq -1$$

```
void mc_integral2()
{
    ran=new TRandom3(37999);
    int N=10000;
    double x,y,z,ma,mx,my,mz,em,emx,emy,emz;
    double sm=0,smx=0,smy=0,smz=0, vm=0,vmx=0,vmy=0,vmz=0;
    double vol=3.*7.*2.;
    for(int i=0; i<N; i++)
    {
        x= 1.+3.*(ran->Rndm());
        y=-3.+7.*(ran->Rndm());
        z=-1.+2.*(ran->Rndm());
        if(z*z+(sqrt(x*x+y*y)-3.)*(sqrt(x*x+y*y)-3.)<1.)
        {
            double den=1.;
            sm+=den; smx+=x*den; smy+=y*den; smz+=z*den;
            vm+=den*den; vmx+=x*den*x*den;
            vmy+=y*den*y*den; vmz+=z*den*z*den;
        }
    }
    ma=vol*sm/N;
    mx=vol*smx/N/ma; my=vol*smy/N/ma; mz=vol*smz/N/ma;
    em=vol*sqrt((vm/N-sm/N*sm/N)/N);
    emx=vol*sqrt((vmx/N-smx/N*smx/N)/N)/ma;
    emy=vol*sqrt((vmy/N-smy/N*smy/N)/N)/ma;
    emz=vol*sqrt((vmz/N-smz/N*smz/N)/N)/ma;
    cout << "mass = " << ma << " +- " << em << endl;
    cout << "xcen = " << mx << " +- " << emx << endl;
    cout << "ycen = " << my << " +- " << emy << endl;
    cout << "zcen = " << mz << " +- " << emz << endl;
}
```

$$\sigma = \sqrt{\langle x^2 \rangle - \langle x \rangle^2}$$

$$\langle x \rangle = \frac{\sum_{i=1}^N x}{N}, \quad \langle x^2 \rangle = \frac{\sum_{i=1}^N x^2}{N}$$

$$er[\langle x \rangle] = \sigma / \sqrt{N}$$

```
mass = 21.80 +- 0.21
xcen = 2.418 +- 0.025
ycen = 0.162 +- 0.025
zcen = 0.005 +- 0.007
```



# MC Ολοκλήρωση: Μείωση διασποράς

## Ολοκλήρωση σε υποδιαστήματα (stratification)

Στην περίπτωση αυτή το υπολογιζόμενο ολοκλήρωμα θα είναι το άθροισμα δύο ή περισσότερων ολοκληρωμάτων και η διασπορά του το άθροισμα των διασπορών.

Για να πετύχουμε καλύτερη διασπορά χρειάζεται να γνωρίζουμε τη συμπεριφορά της συνάρτησης.

Παίρνοντας ίσα υποδιαστήματα, τουλάχιστον στη μονοδιάστατη περίπτωση, δεν κινδυνεύουμε να αυξήσουμε τη διασπορά. Βέβαια δεν είναι σίγουρο ότι θα έχουμε και αξιοσημείωτη βελτίωση.

# MC Ολοκλήρωση: Μείωση διασποράς

## Σημαντική δειγματοληψία (Importance Sampling)

Στην τεχνική λοιπόν αυτή, αλλάζουμε τη μεταβλητή ολοκλήρωσης ώστε να έχουμε ένα ολοκλήρωμα μικρότερης διασποράς

$$I = \int_a^b y(x) dx = \int_a^b \frac{y(x)}{g(x)} g(x) dx = \int_{G(a)}^{G(b)} \frac{y(x)}{g(x)} dG(x) \qquad G(x) = \int_a^x g(x) dx$$

Επομένως πρέπει να βρούμε μία συνάρτηση  $g(x)$  τέτοια ώστε:

- Η  $g(x)$  να είναι συνάρτηση πυκνότητας πιθανότητας, δηλαδή να είναι παντού θετική και κανονικοποιημένη ώστε  $G(b)=1$ .
- Η  $G(x)$  να είναι γνωστή αναλυτικά
- Η  $G(x)$  να μπορεί να λυθεί ως προς  $x$ , ή να υπάρχει γεννήτορας τυχαίων αριθμών που να γεννά σημεία  $x$  σύμφωνα με τη  $g(x)$
- Ο λόγος  $y(x)/g(x)$  να είναι επαρκώς περισσότερο σταθερός από την  $y(x)$  ώστε να μειωθεί η διασπορά

Ασταθές σχήμα αν  $g(x) \ll$

# MC Ολοκλήρωση: Μείωση διασποράς

## Importance Sampling: Ένα παράδειγμα

$$I = \int_0^1 e^{-x^2} dx$$

$$g(x) = ae^{-x} \Rightarrow G(x) = a \int_0^x g(x) dx = a(1 - e^{-x})$$

$$\text{όπου } a = \frac{1}{1 - e^{-1}}$$

Λύνοντας ως προς  $x$ , έχουμε:

$$x = -\log(1 - G/a)$$

Όπου δειγματοληπτούμε ομοιόμορφα τη  $G$ .

Οπότε το ολοκλήρωμα γίνεται:



$$I = \int_0^1 e^{-x^2} dx = \frac{1}{n} \sum_{i=1}^n \frac{e^{-x_i^2}}{e^{-x_i} / a}$$

# MC Ολοκλήρωση: Μείωση διασποράς

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <TRandom.h>

void mc_integ(int N, int nstr)
{
    ran=new TRandom3(17179);
    double sum=0., sum2=0.;
    double sum_is=0., sum2_is=0.;
    double volume = 1.;
    double a=1./(1.-exp(-1.*volume));

    for(int i=0;i<N;i++)
    {
        double x=ran->Rndm()*volume;
        double y=exp(-x*x);
        sum+= y; sum2+=y*y;
// Importance sampling
        x=ran->Rndm();
        double x_is=-log(1.-x/a);
        y=exp(-x_is*x_is)/exp(-x_is)/a;
        sum_is+= y; sum2_is+=y*y;
    }
    double Inte = volume*(sum/N);
    double eInte = volume*sqrt((sum2/N-sum*sum/N/N)/N);
    double Inte_is = sum_is/N;
    double eInte_is = sqrt((sum2_is/N-sum_is*sum_is/N/N)/N);
    cout<<N<<" "<<Inte<<" +- "<<eInte<<" " <<Inte_is<<" +- "<<eInte_is< <endl;
```

# MC Ολοκλήρωση: Μείωση διασποράς

```
// Stratification
const int nn=nstr;
double sum_st[nn]={0.}, sum2_st[nn]={0.};
double h=volume/nstr;
double Inte_st=0;
double eInte_st=0;
for(int j=0; j<nstr; j++)
{
    for(int i=0; i<N/nstr; i++)
    {
        double x=ran->Rndm()*h+j*h;
        double y=exp(-x*x);
        sum_st[j]+= y; sum2_st[j]+=y*y;
    }
}
for(int j=0; j<nstr; j++)
{
    Inte_st +=sum_st[j]*nstr/N;
    eInte_st+=(sum2_st[j]*nstr/N-sum_st[j]*sum_st[j]*nstr*nstr/N/N)*nstr/N;
}
Inte_st=h*Inte_st;
eInte_st=h*sqrt(eInte_st);

cout<<N<<" "<<nstr<<" "<<Inte_st<<" +- "<<eInte_st<<" "<<endl;
}
```

# MC Ολοκλήρωση: Μείωση διασποράς

Πραγματική τιμή : 0.74682

## Importance sampling: Σύγκριση

n	$I_{\text{flat}}$	$\Delta I_{\text{flat}}$	$I_{\text{exp}}$	$\Delta I_{\text{exp}}$
$10^2$	0.76216	0.01880	0.75082	0.00589
$10^3$	0.74683	0.00612	0.75151	0.00172
$10^4$	0.74562	0.00199	0.74800	0.00055
$10^5$	0.74766	0.00063	0.74690	0.00017

## Stratification: Σύγκριση

# διαστημ.	1	2	4	8
n=10 <sup>4</sup>	0.00199	0.00097	0.00049	0.00025

# MC Ολοκλήρωση: Μείωση διασποράς

## Control Variates

$$I = \int y(x) dx = \int [y(x) - g(x)] dx + \int g(x) dx$$

Ευσταθές σχήμα και για  $g(x) \ll$

## Antithetic Variates

$$V[y_1(x) + y_2(x)] = V[y_1(x)] + V[y_2(x)] + 2\text{cov}[y_1(x), y_2(x)]$$

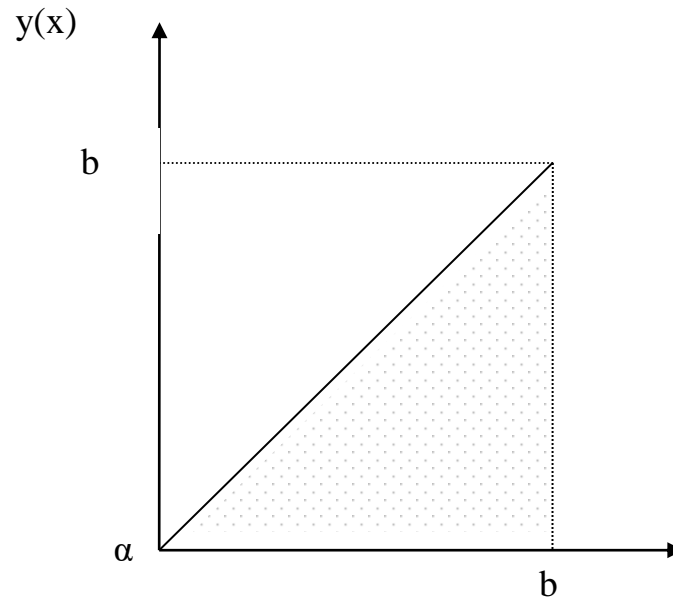
$$I = \int y(x) dx = \int [y_1(x) + y_2(x)] dx$$

$$y_1 = \frac{1}{2} y(x) \quad y_2 = \frac{1}{2} y(b - (x - a))$$

Για μονοτονικές  $y(x)$

# MC Ολοκλήρωση: Εφαρμογή

$$I = \int_a^b dx \int_a^x dy g(x, y)$$



*I) Ο προφανής τρόπος:*

A) Επιλέγουμε  $x_i = R[\alpha, b]$

B) Επιλέγουμε  $y_i = R[\alpha, x_i]$

Γ) Αθροίζουμε τα  $g(x_i, y_i)$  ώστε τελικά να έχουμε:

$$I = \frac{(b-a)^2}{2n} \sum_{i=1}^n g(x_i, y_i)$$

Παρόλο που αυτή η μέθοδος φαίνεται προφανής, είναι **λανθασμένη!** Ο λόγος είναι ότι τα σημεία  $(x_i, y_i)$  δεν είναι ομοιόμορφα καταναμεμένα σε όλη την περιοχή της ολοκλήρωσης.

Υπάρχει περίπου ο ίδιος αριθμός σημείων στην περιοχή  $\alpha < x < (\alpha+b)/2$  όπως και στην περιοχή  $(\alpha+b)/2 < x < b$  παρόλο που η δεύτερη είναι τριπλάσια σε μέγεθος.



# MC Ολοκλήρωση: Εφαρμογή

## II) Η μέθοδος της απόρριψης:

A) Επιλέγουμε  $x_i = R[a,b]$  και  $y_i = R[a,b]$

B) Δημιουργούμε μια νέα συνάρτηση  $z(x,y)$ , η οποία ορίζεται σε όλο το δισδιάστατο χώρο στον οποίο γεννάμε τους τυχαίους αριθμούς, αλλά έχει το ίδιο ολοκλήρωμα με τη  $g(x,y)$  ως:

$$\begin{aligned} z(x_i, y_i) &= 0 && \text{αν } x_i < y_i \\ &= g(x_i, y_i) && \text{αν } x_i > y_i \end{aligned}$$

Το ολοκλήρωμα σ' αυτήν την περίπτωση γίνεται:

$$I = \frac{(b-a)^2}{n} \sum_{i=1}^n z(x_i, y_i)$$

# MC Ολοκλήρωση: Εφαρμογή

## III) Η μέθοδος της απόρριψης (με γνωστή την περιοχή ολοκλήρωσης) :

Ένας άλλος τρόπος θα ήταν να υπολογίσουμε το λόγο  $r$  του χώρου που καταλαμβάνει η περιοχή ολοκλήρωσης σε σχέση με την περιοχή δειγματοληψίας. Αν γνωρίζαμε το λόγο αυτό, θα μπορούσαμε να περιορίσουμε το σφάλμα, απορρίπτοντας τα σημεία που βρίσκονται εκτός της περιοχής ολοκλήρωσης. Δηλαδή, θα ακολουθούσαμε τα εξής βήματα:

A) Επιλέγουμε  $x_i=R[a,b]$  και  $y_i=R[a,b]$

B) Απορρίπτουμε το σημείο αν δεν ανήκει στην περιοχή ολοκλήρωσης, δηλαδή, αν  $y_i > x_i$ .

Γ) Αθροίζουμε τα  $g(x_i, y_i)$ , αντικαθιστώντας τον όγκο δειγματοληψίας με τον όγκο ολοκλήρωσης  $r(a-b)^2$ . Στο συγκεκριμένο παράδειγμα γνωρίζουμε ότι  $r=1/2$  άρα:

Το ολοκλήρωμα σ' αυτήν την περίπτωση γίνεται:

$$I = \frac{1}{2} \frac{(b-a)^2}{n'} \sum_{i=1}^n g(x_i, y_i)$$

# MC Ολοκλήρωση: Εφαρμογή

## *IV) Η μέθοδος της αναδίπλωσης (ένα τρικ)*

A) Επιλέγουμε  $u_i = R[a, b]$  και  $v_i = R[a, b]$

B) Θέτουμε  $x_i = \max(u_i, v_i)$  και  $y_i = \min(u_i, v_i)$

Το ολοκλήρωμα δίνεται από:

$$I = \frac{1}{2} \frac{(b-a)^2}{n} \sum_{i=1}^n g(x_i, y_i)$$

Αυτή η μεθοδολογία ισοδυναμεί με την ομοιόμορφη δειγματοληψία όλου του τετραγώνου και στη συνέχεια την αναδίπλωσή του γύρω από τη διαγώνιο, έτσι ώστε όλα τα σημεία να πέσουν στην περιοχή της ολοκλήρωσης. Η πυκνότητα των σημείων παραμένει ομοιόμορφη.

# MC Ολοκλήρωση: Εφαρμογή

## V) Η μέθοδος βάρους

Τα πρώτα δύο βήματα είναι ίδια με την πρώτη (λανθασμένη) μέθοδο που αναφέραμε.

A) Επιλέγουμε  $x_i = R[a, b]$

B) Επιλέγουμε  $y_i = R[a, x_i]$

Γ) Αλλά, στο τρίτο βήμα πραγματοποιούμε το άθροισμα με βάρος (κάποιο συντελεστή). Το βάρος διορθώνει την ανομοιόμορφη πυκνότητα σημείων (στο παράδειγμα αυτό  $1/(x-a)$ ). Επομένως το ολοκλήρωμα δίνεται από:

$$I = \frac{(b-a)}{n} \sum_{i=1}^n (x_i - a) g(x_i, y_i)$$

Η μέθοδος αυτή είναι πρακτικά μια εφαρμογή της μεθόδου σημαντικής δειγματοληψίας. Μπορεί να είναι πιο αποδοτική ή όχι από τη μέθοδο αναδίπλωσης, ανάλογα με το αν η  $(x-a)g$  ή η  $g$  έχει μικρότερη διασπορά.

# Προσομοίωση Monte Carlo

## Πως υλοποιείται:

- Επιλογή της συνάρτησης πυκνότητας πιθανότητας (pdf) του υποθετικού πληθυσμού.
- Γέννηση τυχαίου δείγματος από μια ακολουθία ψευδοτυχαίων αριθμών σύμφωνα με την παραπάνω πιθανότητα.
- Στατιστική επεξεργασία για την εκτίμηση των υπό εξέταση παραμέτρων από το τυχαίο δείγμα.

# Προσομοίωση MC: Τεχνικό παράδειγμα

Έστω θέλουμε  $f(x)=1+x^2$  στην περιοχή  $[-1.0,+1.0]$

## I) Μέθοδος βάρους

Η διαδικασία στη μέθοδο αυτή είναι μάλλον τετριμμένη. Γεννάμε ομοιόμορφους τυχαίους αριθμούς  $x_i=R[-1.0,+1.0]$ , και τους επισυνάπτουμε βάρος  $w_i=1+x_i^2$ .

## II) Μέθοδος απόρριψης

Στην περίπτωση αυτή έχουμε  $f_{\max}=2.0$  στο διάστημά μας.

Άρα γεννάμε:

$$x_i = R[-1.,+1.] = 2R[0.,1.]-1. \text{ και}$$

$$r_i = R[0.,+2.] = 2R[0.,1.]$$

Απορρίπτουμε το σημείο αν  $r_i > 1.+ x_i^2$

Να σημειώσουμε εδώ ότι η απόδοση της προσομοίωσης στην περίπτωση αυτή είναι:

$$\varepsilon = \frac{\int_{-1}^1 (1+x^2)dx}{[1.-(-1.)]f_{\max}} = \frac{2}{3}$$

με άλλα λόγια το 1/3 των σημείων που γεννάμε απορρίπτεται.

# Προσομοίωση MC: Τεχνικό παράδειγμα

## III) Μέθοδος του αντιστρόφου μετασχηματισμού

Η μέθοδος του αντιστρόφου μετασχηματισμού είναι η αντίστοιχη της σημαντικής δειγματοληψίας στην ολοκλήρωση. Μετασχηματίζουμε την  $f(x)$  στην  $F(x)$  με  $f(x)dx=dF$ , οπότε η  $F(x)$  δεν είναι απλά η αθροιστική πυκνότητα πιθανότητας

$$F(x) = \int_{x_{\min}}^x f(x)dx \quad \Rightarrow \quad F(x) = \int_{-1}^x (1+x^2)dx = x + \frac{x^3}{3} + \frac{4}{3}$$

επομένως  $F(-1)=0$ . και  $F(+1.)=8/3$ . Ως εκ τούτου, γεννώντας  $u_i=R[0.,1.]$ , το  $8u_i/3$  κατανέμεται ομοιόμορφα στο  $[F(-1.),F(+1.)]$ . Η τιμή του  $x$  που αντιστοιχεί είναι η λύση της:

$$\frac{8}{3}u = F(x) = x + \frac{x^3}{3} + \frac{4}{3} \quad \Rightarrow \quad x_i=A+B \text{ με } \begin{cases} A = (4u_i - 2 + s)^{1/3} \\ B = (4u_i - 2 - s)^{1/3} \\ s = \sqrt{1 + 4(1 - 2u_i)^2} \end{cases}$$

Με τη μέθοδο του αντίστροφου μετασχηματισμού έχουμε απόδοση 1, αλλά για κάθε σημείο απαιτείται ο υπολογισμός δύο κυβικών και μίας τετραγωνικής ρίζας. Αξίζει λοιπόν, να πειραματιστεί κανείς για να δει αν τελικά έχει τελικά κάποιο υπολογιστικό όφελος σε σχέση με τη μέθοδο της απόρριψης.

# Προσομοίωση MC: Τεχνικό παράδειγμα

## IV) Η σύνθετη μέθοδος

Εκφράζουμε την  $f(x)$  ως άθροισμα απλούστερων συναρτήσεων.

Στην περίπτωση του παραδείγματος που εξετάζουμε, η επιλογή είναι μάλλον προφανής:

$$f(x) = f_a(x) + f_b(x) \quad \text{με } f_a(x) = 1. \text{ και } f_b(x) = x^2.$$

Τα ολοκληρώματα των δύο επιμέρους συναρτήσεων είναι:

$$\int_{-1}^{+1} f_a(x) dx = 2 \qquad \int_{-1}^{+1} f_b(x) dx = \frac{2}{3}$$

Επομένως θέλουμε να γεννάμε σημεία από την  $f_a(x)$  και  $f_b(x)$  με πιθανότητα  $3/4$  και  $1/4$  αντίστοιχα. Ακολουθούμε λοιπόν την εξής πορεία:

Γεννάμε ομοιόμορφα κατανομημένο τυχαίο  $v_i = R[0., 1.]$

➤ Αν  $v_i < 3/4$  γεννάμε από την  $f_a$ :  $u_i = R[0., 1.]$  και  $x_i = 2u_i - 1$

➤ Αν  $v_i > 3/4$  γεννάμε από την  $f_b$ :

Είτε με τη μέθοδο της απόρριψης (συνδυασμένη απόδοση 5/6)

Είτε με τη μέθοδο του μετασχηματισμού (μία μόνο κυβική ρίζα)



Μελετάμε με απλό τρόπο την πορεία ενός συστήματος προς την ισορροπία. Έστω ένα κουτί που χωρίζεται στη μέση από ένα διαχωριστικό πέτασμα και περιέχει μόρια ενός αερίου (σωματίδια).

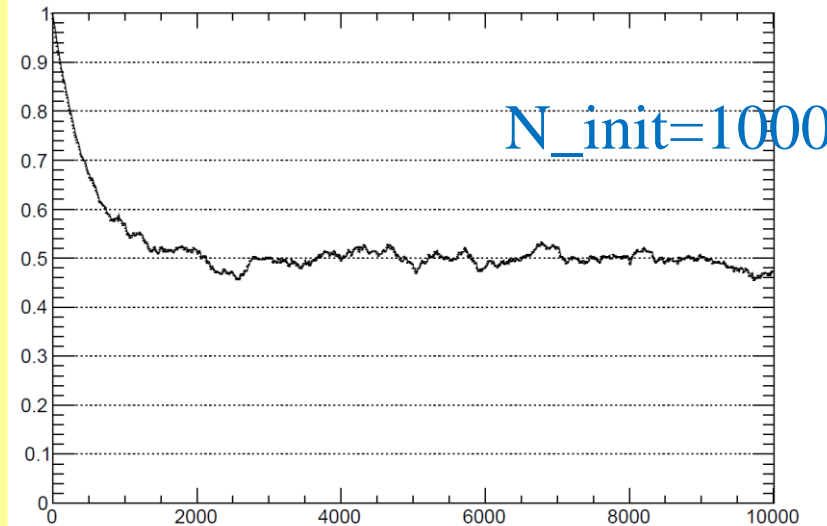
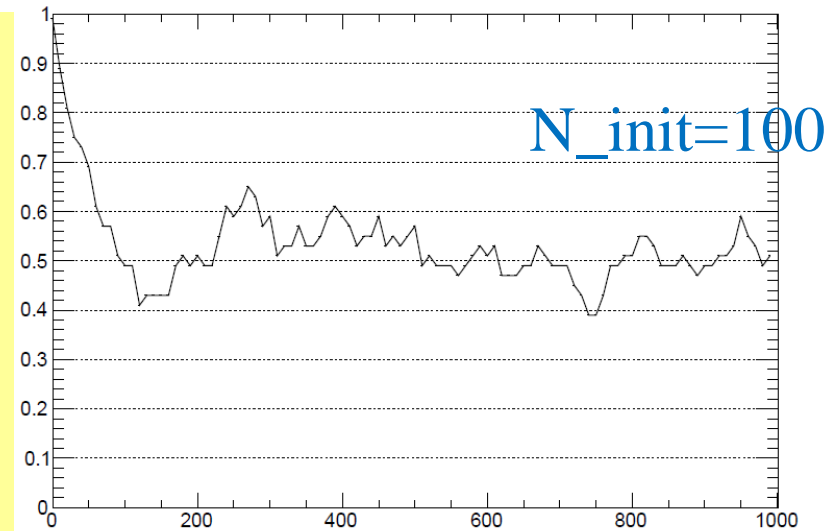
Τη χρονική στιγμή  $t=0$ , όλα τα μόρια βρίσκονται στο αριστερό μέρος του κουτιού, ενώ ανοίγεται μια τρύπα στο πέτασμα η οποία επιτρέπει τη διέλευση ενός σωματιδίου σε χρονικό διάστημα  $dt$ .

Δεδομένου ότι θεωρούμε πολλά τέτοια σωματίδια μπορούμε να μελετήσουμε την πορεία του συστήματος προς την ισορροπία, χωρίς να εξετάζουμε τις αρχικές συνθήκες και την κίνηση κάθε σωματιδίου ξεχωριστά.

Θεωρούμε ότι όλα τα σωματίδια στο αριστερό μέρος του κουτιού  $N_{\text{left}}$  έχουν την ίδια πιθανότητα να διέλθουν προς τα δεξιά και ως εκ τούτου η πιθανότητα για να συμβεί αυτό θα είναι  $N_{\text{left}}/N$

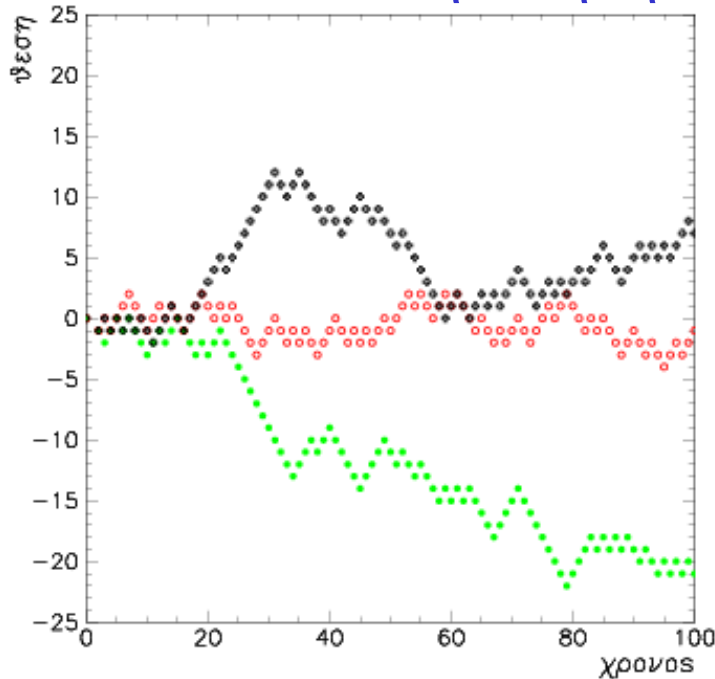
# ΥΠΟΛΟΓΙΣΤΙΚΗ ΦΥΣΙΚΗ - MC Προσομοίωση: Σωματίδια σε κουτί

```
#include <iostream>
#include <math>
#include "TRandom3.h"
void par_in_box(const int N_init)
{
    int Nbins=N_init;
    ran=new TRandom3(779977);
    int max_time=10*N_init;
    int N_left=N_init;
    float time[N_init];
    float pososto[N_init];
    for(int t=0; t<max_time; t++)
    {
        int x= N_init*ran->Rndm();
        if(x<N_left)
            {N_left--;} else{N_left++;}
        if(t%10==0){
            time[t/10]=t;
            pososto[t/10]=(float)N_left/N_init;
        }
    }
    TGraph *gr = new TGraph(N_init,time,pososto);
    gr->Draw();
}
```



# Προσομοίωση MC: Τυχαία κίνηση

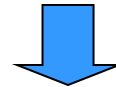
## Μονοδιάστατη κίνηση



```

SUBROUTINE RANDOM_WALK1(ISEED)
* A subroutine for 1 dimensional random walk simulation
* with unit step
  IDUMMY=ISEED ! Initialisation of random sequence
  NSTEP=100    ! number of time steps
  X = 0.       ! Initial position
  T = 0.       ! at time = 0
  DO ISTEP=2,NSTEP ! Perform the time steps
    R = RUN0(IDUMMY)-0.5 ! random number in -0.5 - 0.5
    X = X+ABS(R)/R      ! move +1 or -1 according to sign of R
*    X = X+2.*R         ! move uniformly in -1 : +1 range
    T = ISTEP          ! time according to steps
  ENDDO
99  CONTINUE
  RETURN
  END
    
```

$$x_n = \sum_{i=1}^n r_i \quad x_n^2 = \sum_{i=1}^n \left( \sum_{j=1}^n r_i r_j \right)$$



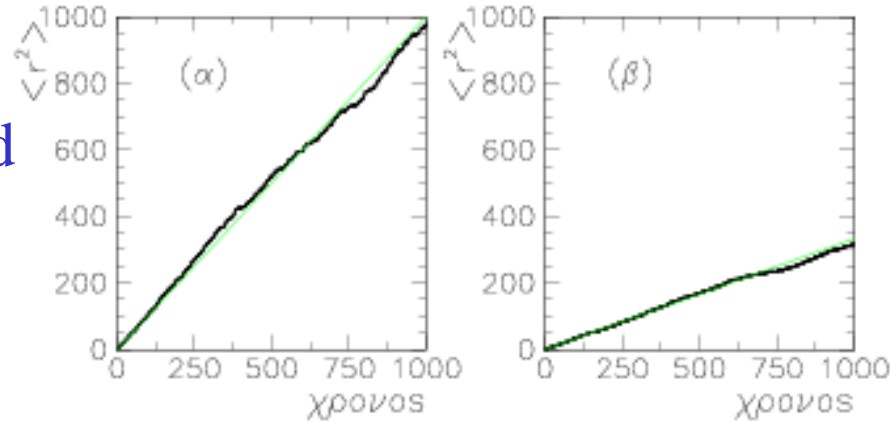
$$\langle x_n^2 \rangle = \infty t \quad \leftarrow \quad \langle x_n \rangle = 0 \quad \langle x_n^2 \rangle = \sum_{i=1}^n r_i^2 = n \langle r_i^2 \rangle$$

D

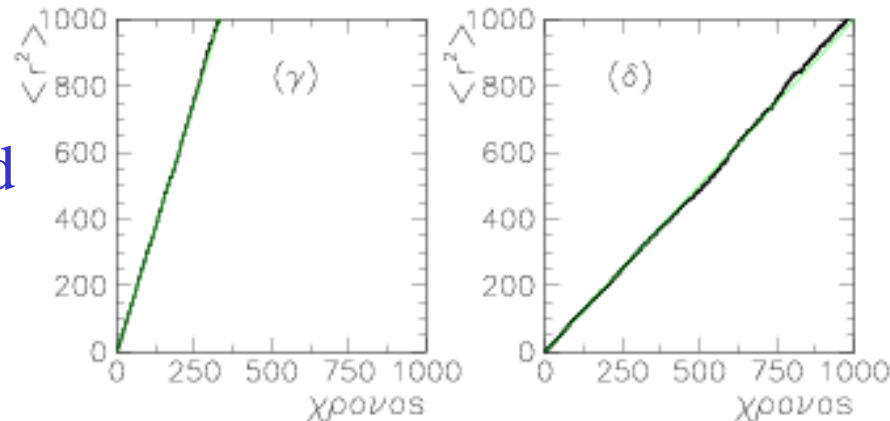
# Προσομοίωση MC: Τυχαία κίνηση

## Παραδείγματα διαφόρων D

1d



3d

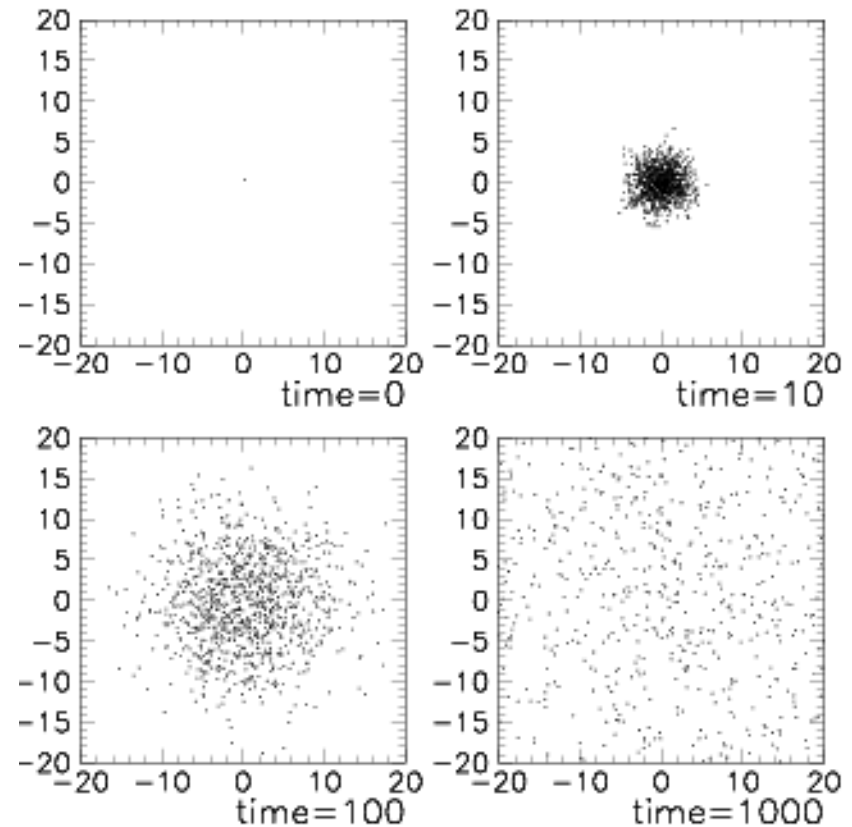
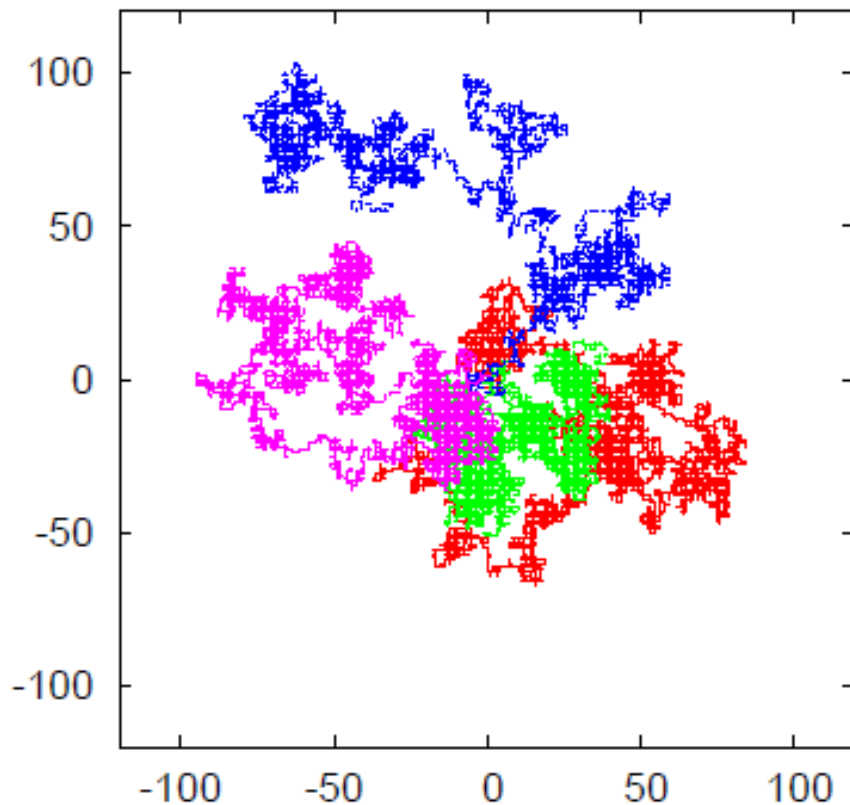


Βήμα 1.

Βήμα 0.-1.

# Προσομοίωση MC: Τυχαία κίνηση

## Παράδειγμα σε 2d



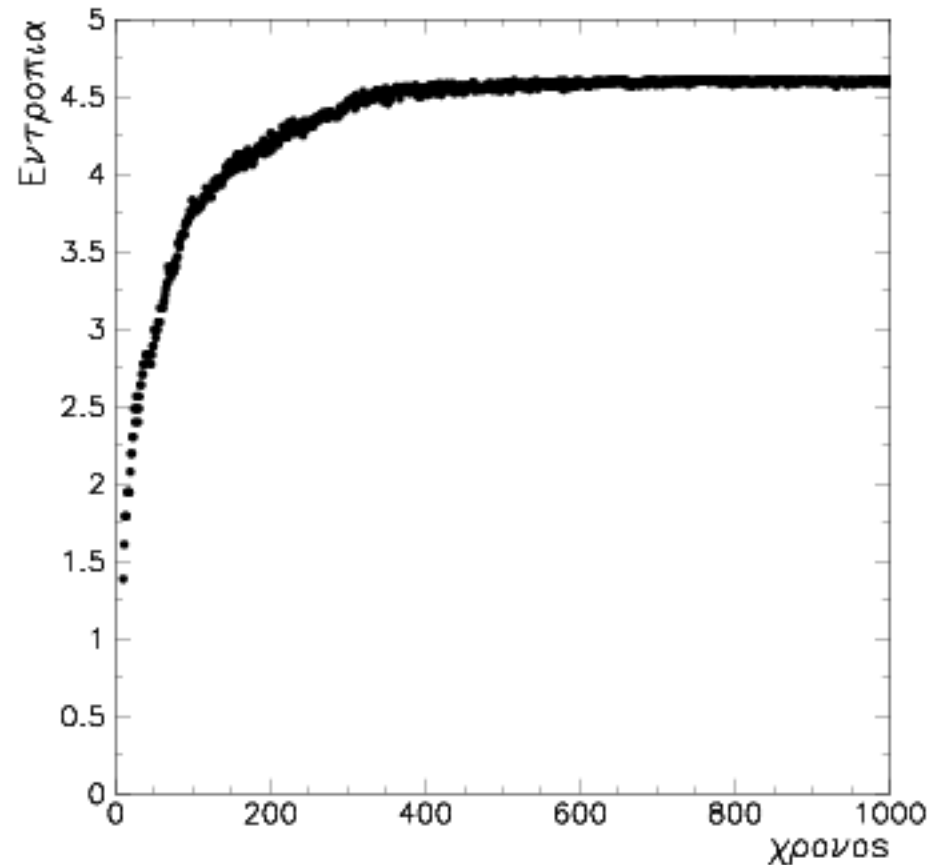
Αύξηση της εντροπίας λόγω  
τυχειότητας της κίνησης

# Προσομοίωση MC: Τυχαία κίνηση

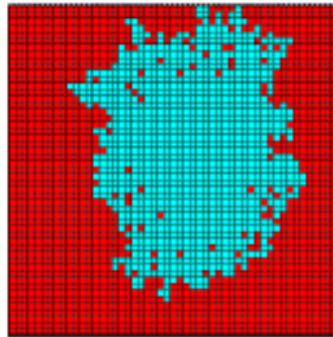
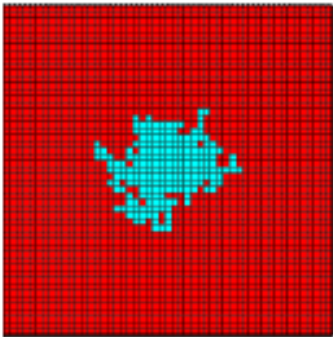
## Αύξηση Εντροπίας

$$S = -\sum_i P_i \ln P_i$$

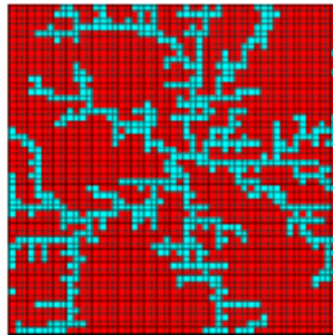
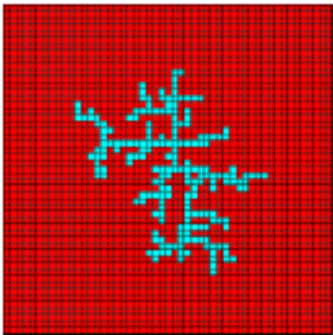
Χωρίζουμε το χώρο μας σε κομμάτια ίσων εμβαδών και υπολογίζουμε με την προσομοίωσή μας την πιθανότητα κάθε ένα από αυτά τα κομμάτια να πληρώνεται από τυχαίους περιπατητές.



# Προσομοίωση MC: Cluster growth



Eden model  
 $m(r) \sim r^{df}$   $df \sim 2$



DLA model  
 $m(r) \sim r^{df}$   $df \sim ?$

df=fractal dimensionality

# Προσομοίωση MC: Cluster growth 1

## **SUBROUTINE MC\_CLUSTER1 (IDUMMY)**

```
* Cluster growth routine according
* to Eden model (cancer model)
  IMPLICIT NONE
  INTEGER NGRID,NWALK
  PARAMETER (NGRID=101,NWALK=1000)
  REAL AGRID(NGRID,NGRID)
  INTEGER IDUMMY
  INTEGER I,J,IGRX,IGRY
  EXTERNAL RUN0, NEIGHBOR

* Initialise grid
  DO I=1,NGRID
    DO J=1,NGRID
      AGRID(I,J)=0
    ENDDO
  ENDDO

* Initial particle at the centre
  IGRX=INT(NGRID/2)+1
  IGRY=INT(NGRID/2)+1
  AGRID(IGRX,IGRY)=1.

* Loop over the walkers
  DO 100 I=1,NWALK

* Choose a position close to neighbours
  DO WHILE (NEIGHBOR(AGRID,IGRX,IGRY).EQ.0)
    IGRX=INT(NGRID*RUN0(IDUMMY))+1
    IGRY=INT(NGRID*RUN0(IDUMMY))+1
  ENDDO
  AGRID(IGRX,IGRY)=1.

100 CONTINUE
```

\*Output cluster configuration

```
DO I =1, NGRID
  DO J=1,NGRID
    CALL HFILL(300,FLOAT(I),FLOAT(J),AGRID(I,J))
  ENDDO
ENDDO
RETURN
END

FUNCTION NEIGHBOR(AGRID,IGRX,IGRY)
  INTEGER NEIGHBOR
  INTEGER IGRX,IGRY
  INTEGER IGX1,IGX2,IGY1,IGY2
  INTEGER NGRID
  PARAMETER (NGRID=101)
  REAL AGRID(NGRID,NGRID)
  NEIGHBOR = 0
  IF (AGRID(IGRX,IGRY).GT.0.5) RETURN
  IGX1=IGRX-1
  IGX2=IGRX+1
  IGY1=IGRY-1
  IGY2=IGRY+1
  IF (IGRX.EQ.1) IGX1=NGRID
  IF (IGRX.EQ.NGRID) IGX2=1
  IF (IGRY.EQ.1) IGY1=NGRID
  IF (IGRY.EQ.NGRID) IGY2=1
  IF (AGRID(IGX1,IGRY).GT.0.5.OR.
&AGRID(IGX2,IGRY).GT.0.5.OR.
&AGRID(IGRX,IGY1).GT.0.5.OR.
&AGRID(IGRX,IGY2).GT.0.5)
    NEIGHBOR = 1
  RETURN
END
```



## Προσομοίωση MC: Cluster growth 2

```
SUBROUTINE MC_CLUSTER2(IDUMMY)
```

•Cluster growth routine according to

\*DLA model (snowflake model)

```
IMPLICIT NONE
```

```
INTEGER NGRID,NWALK
```

```
PARAMETER(NGRID=101,NWALK=1000)
```

```
REAL AGRID(NGRID,NGRID)
```

```
INTEGER IDUMMY
```

```
INTEGER I,J,IGRX,IGRY
```

```
EXTERNAL RUN0, NEIGHBOR
```

\* Initialize random numbers

```
CALL RDMIN(ISEED)
```

\* Initialize grid

```
DO I=1,NGRID
```

```
DO J=1,NGRID
```

```
AGRID(I,J)=0
```

```
ENDDO
```

```
ENDDO
```

\* Initial particle at the center

```
IGRX=INT(NGRID/2)+1
```

```
IGRY=INT(NGRID/2)+1
```

```
AGRID(IGRX,IGRY)=1.
```

\* Loop over the walkers

```
DO 100 I=1,NWALK
```

•Choose initial walker position

\*at some corner

```
IG=INT(4.*RNDM(IDUMMY))
```

```
IF(IG.EQ.0) THEN
```

```
IGRX=1
```

```
IGRY=1
```

```
ELSEIF(IG.EQ.1) THEN
```

```
IGRX=NGRID
```

```
IGRY=1
```

```
ELSEIF(IG.EQ.2) THEN
```

```
IGRX=1
```

```
IGRY=NGRID
```

```
ELSEIF(IG.EQ.3) THEN
```

```
IGRX=NGRID
```

```
IGRY=NGRID
```

```
ENDIF
```

\* Start random walk

```
DO WHILE(NEIGHBOR(AGRID,IGRX,IGRY).EQ.0)
```

```
IG=INT(4.*RNDM(IDUMMY))
```

```
IF(IG.EQ.0.AND.IGRX.NE.1) IGRX=IGRX-1
```

```
IF(IG.EQ.1.AND.IGRX.NE.NGRID) IGRX=IGRX+1
```

```
IF(IG.EQ.2.AND.IGRY.NE.1) IGRY=IGRY-1
```

```
IF(IG.EQ.3.AND.IGRY.NE.NGRID) IGRY=IGRY+1
```

```
ENDDO
```

```
AGRID(IGRX,IGRY)=1.
```

```
100 CONTINUE
```

\* Output cluster configuration

```
DO I =1, NGRID
```

```
DO J=1,NGRID
```

```
CALL HFILL(400,FLOAT(I),FLOAT(J),AGRID(I,J))
```

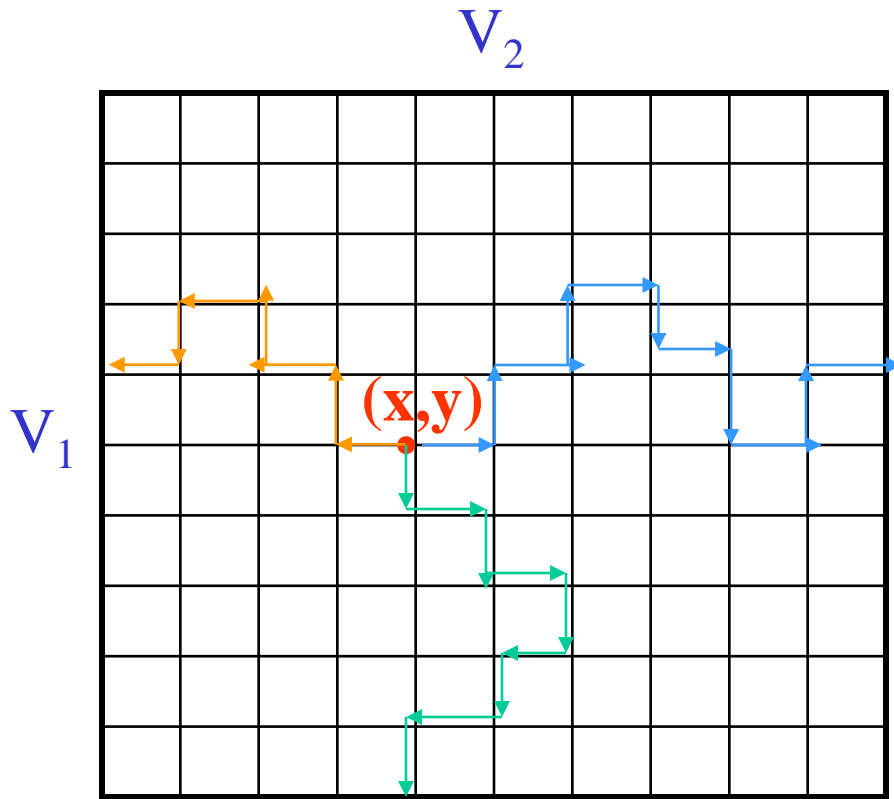
```
ENDDO
```

```
ENDDO
```

```
RETURN
```

```
END
```

# MC: Επίλυση προβλημάτων ευρέσεως δυναμικού



Χρήση τυχαίας κίνησης σταθερού βήματος

$$V(x, y) = \frac{1}{N} \sum_{i=1}^N V_b(i)$$

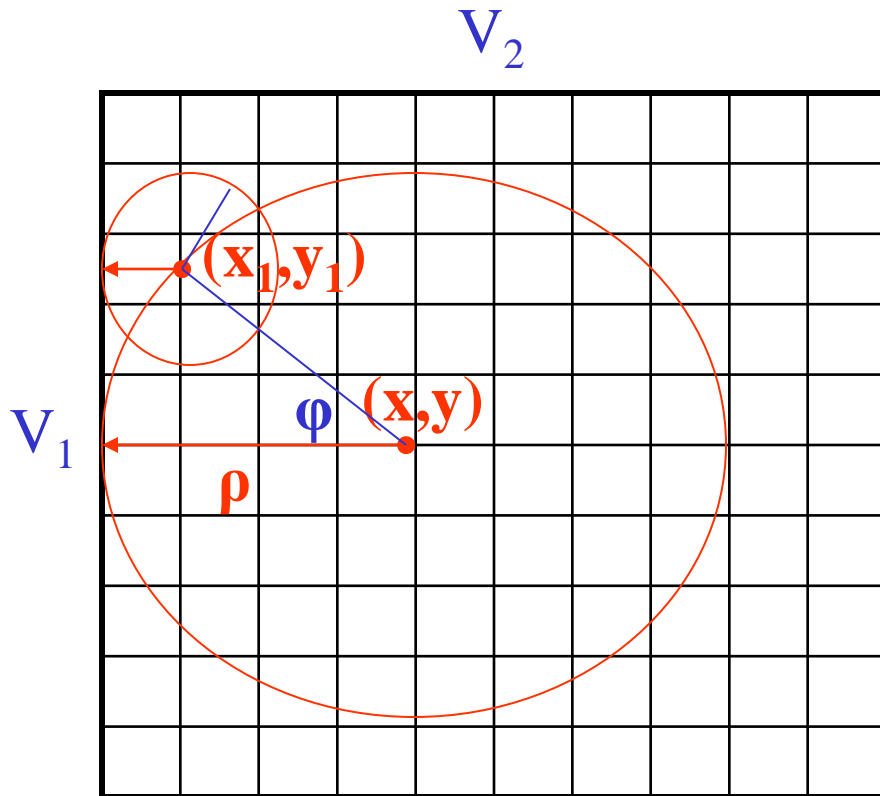
$N$  τυχαίες κινήσεις

$$\nabla^2 V = 0$$

$$V(x, y) = p_{x+} V(x + h, y) + p_{x-} V(x - h, y) + p_{y+} V(x, y + h) + p_{y-} V(x, y - h)$$

Όπου  $p_i = 1/4$

# MC: Επίλυση προβλημάτων ευρέσεως δυναμικού



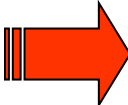
Χρήση τυχαίας κίνησης  
μεταβαλλόμενου βήματος

Τυχαία κίνηση:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \rho_i \cos \phi_i$$

$$\mathbf{y}_{i+1} = \mathbf{y}_i + \rho_i \sin \phi_i$$

μέχρι να φτάσουμε  
πολύ κοντά σε σύνορο

$$\nabla^2 V = 0$$


$$V(x, y) = \frac{1}{N} \sum_{i=1}^N V_b(i)$$

**N** τυχαίες κινήσεις

Σχεδόν το 90% από τους 2500 γνωστούς πυρήνες είναι ραδιενεργοί, δηλαδή δεν είναι σταθεροί και διασπώνται.

Έστω ότι έχουμε ένα πληθυσμό από ραδιενεργούς πυρήνες  $N(t)$ .

Αν κανονικοποιήσουμε ως προς τον αρχικό πληθυσμό  $N(0)=N_0$ , τότε

$N(t)$  εκφράζει την πιθανότητα ενός πυρήνα να μην έχει διασπαστεί σε χρόνο  $t$

Έστω  $P(t)=-dN/dt$  η πιθανότητα ενός πυρήνα να διασπαστεί τη χρονική στιγμή  $t$

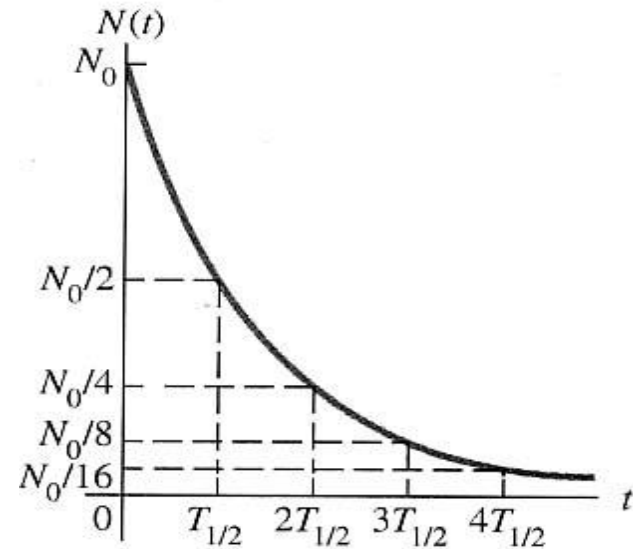
ΟΠΟΤΕ

$$P(t) = \lambda N(t) \rightarrow N(t) = \exp(-\lambda t)$$

Ενεργότητα  $\frac{dN}{dt} = -\lambda N \Rightarrow N = N_0 e^{-\lambda t}$

Χρόνος ημιζωής  $T_{1/2}$

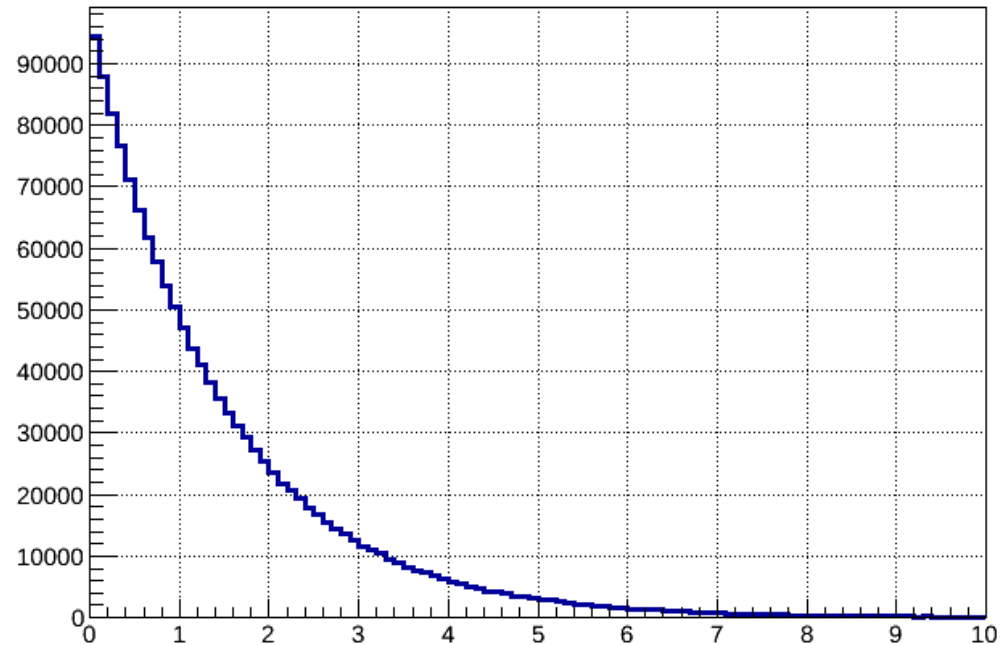
$$N = \frac{N_0}{2} = N_0 e^{-\lambda T_{1/2}} \Rightarrow T_{1/2} = \frac{\ln 2}{\lambda} = \frac{0.693}{\lambda}$$



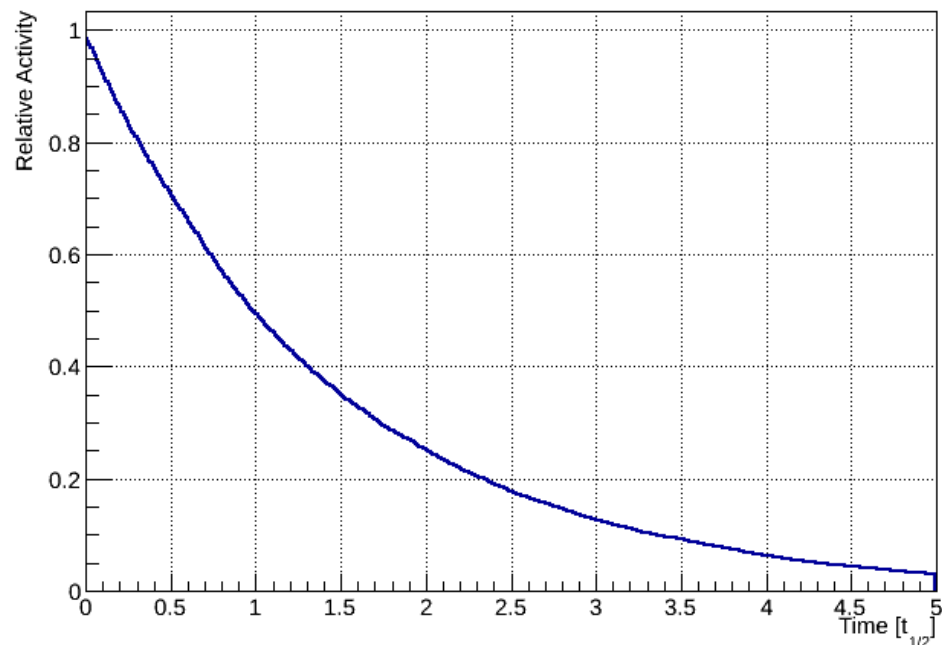
$$p(y)dy = \left| \frac{dx}{dy} \right| dy = e^{-\lambda y} dy$$

$$p(y)dy = \left| \frac{dx}{dy} \right| dy = e^{-\lambda y} dy \Rightarrow x = e^{-\lambda y} \Rightarrow y = -\ln(x) / \lambda$$

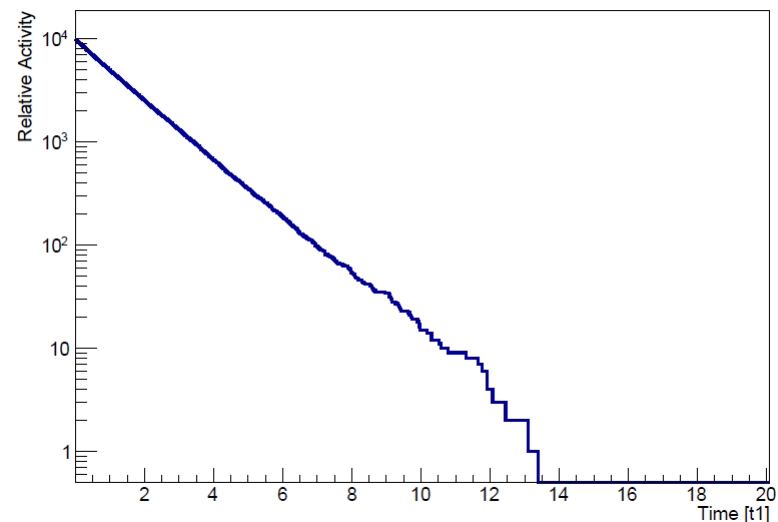
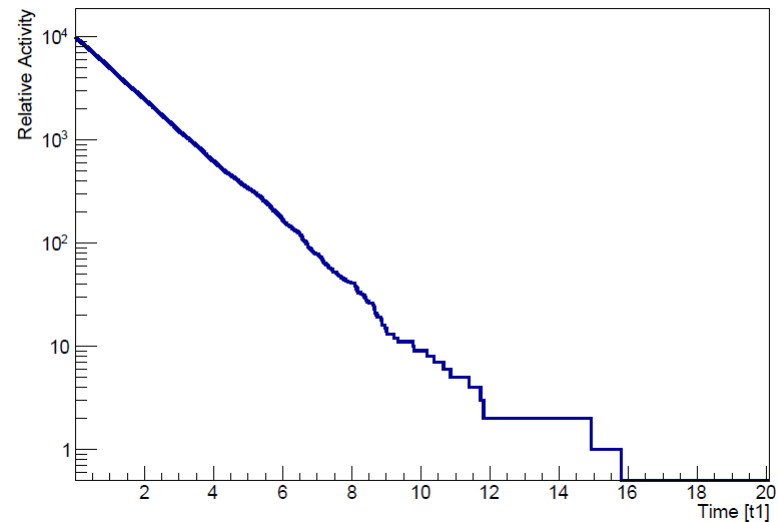
```
r=-log(ran->Rndm())/lamda;
```



```
void nuclear_decays(double t1)
{
    ran=new TRandom3(1717);
    TH1F *hc1=new TH1F("hc1"," ",500, 0., 5.);
    double r,dt;
    double lamda=log(2.)/t1;
    dt=t1/100.;
    float N0=10000.;
    float N1=N0;
    for(int i=0;i<500;i++)
    {
        for(int il=0;il<N1;il++)
        {
            r=-log(ran->Rndm())/lamda;
            if(r<=dt)N1--;
        }
        hc1->SetBinContent(i,N1/N0);
        // printf("%f \n ",N1/N0);
    }
    hc1->Draw();
}
```



```
void nuclear_decays(double t1)
{
    ran=new TRandom3(1717);
    TH1F *hc1=new TH1F("hc1"," ",500, 0., 5.);
    double r,dt;
    double lamda=log(2.)/t1;
    dt=t1/100.;
    float N0=10000.;
    float N1=N0;
    for(int i=0;i<500;i++)
    {
        for(int il=0;il<N1;il++)
        {
            r=-log(ran->Rndm())/lamda;
            if(r<=dt)N1--;
        }
        hc1->SetBinContent(i,N1/N0);
        // printf("%f \n ",N1/N0);
    }
    hc1->Draw();
}
```

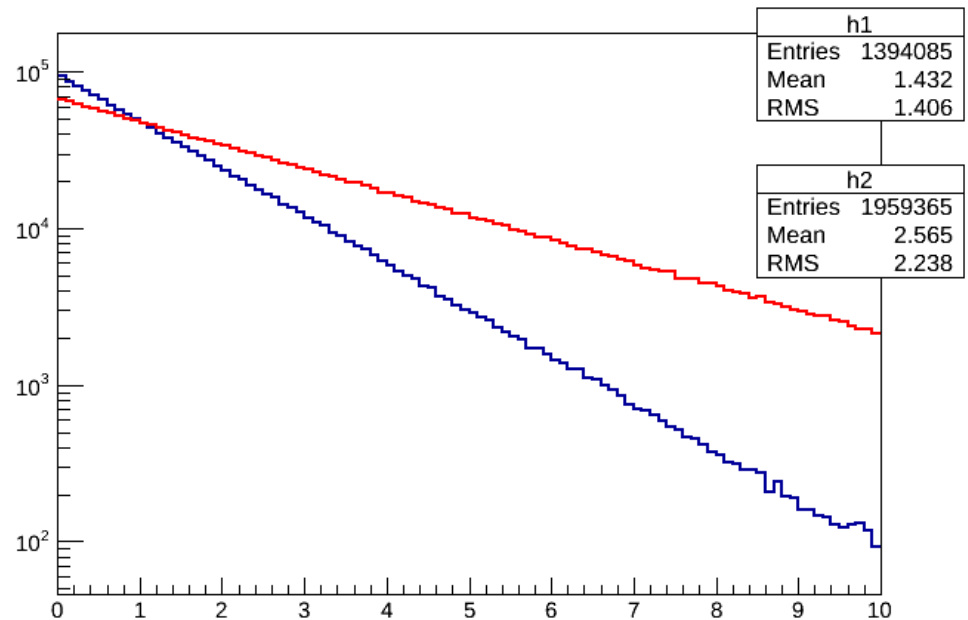


Για αυτό όμως δε χρειαζόταν προσομοίωση MC, αφού έχουμε την αναλυτική λύση!

Έστω όμως ότι έχουμε ένα πληθυσμό από ραδιενεργούς πυρήνες, οι οποίοι διασπώνται σε επίσης ραδιενεργούς πυρήνες με διαφορετικός χρόνο ημιζωής. Πόση θα είναι η ενεργότητα κάθε είδους πυρήνα, ως συνάρτηση του χρόνου;

```
r1=-log(ran->Rndm())/l1;  
r2=-log(ran->Rndm())/l2;
```

```
N1--;    →    N2++;
```



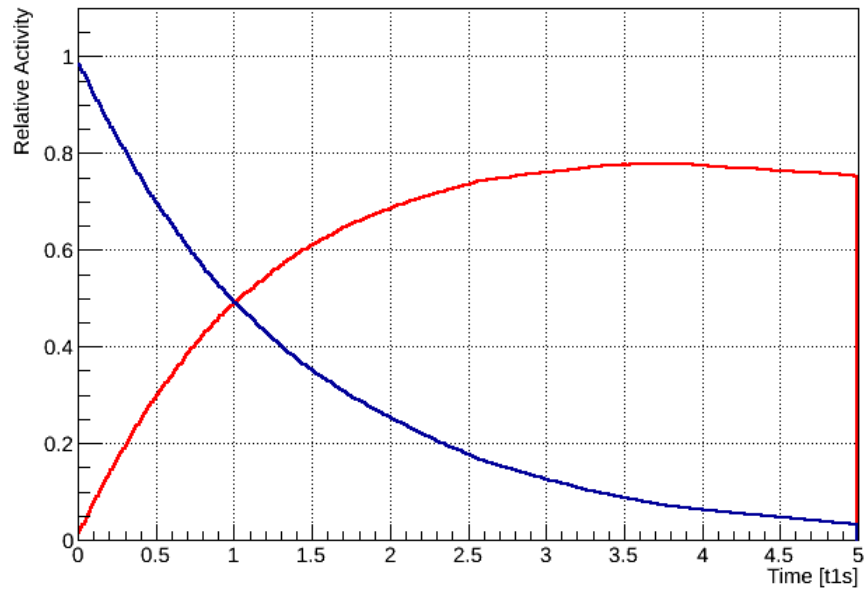


```

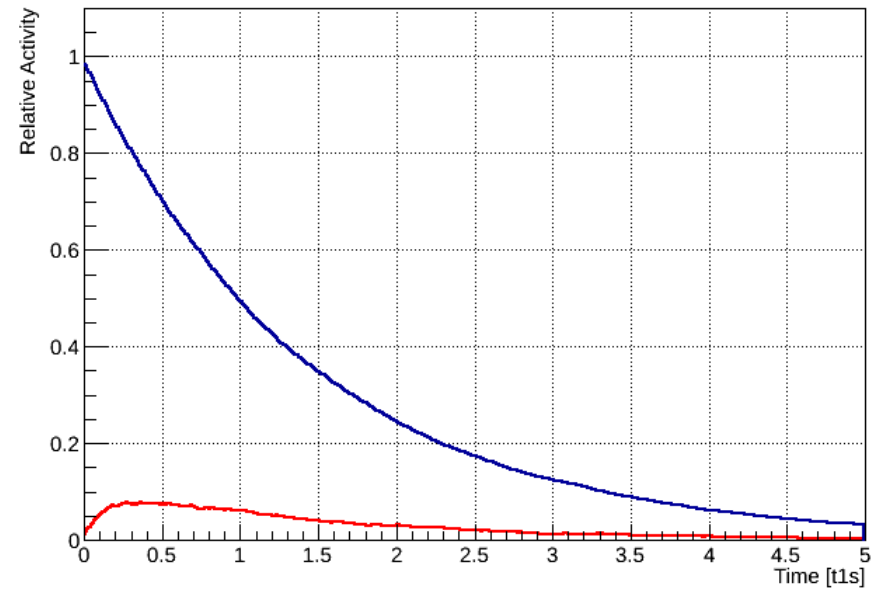
void nuclear_decays(double t1, double t2)
{
    ran=new TRandom3(1717);
    TH1F *hc1=new TH1F("hc1"," ",500, 0., 5.);
    TH1F *hc2=new TH1F("hc2"," ",500, 0., 5.);
    double r,dt;
    double l1=log(2.)/t1; double l2=log(2.)/t2;
    dt=t1/100.;
    float N0=10000.;
    float N1=N0, N2=0.;
    for(int i=0;i<500;i++)
    {
        for(int i1=0;i1<N1; i1++)
        {
            r=-log(ran->Rndm())/l1;
            if(r<=dt){ N1--; N2++; }
        }
        for(int i2=0; i2<N2; i2++)
        {
            r=-log(ran->Rndm())/l2;
            if(r<=dt)N2--;
        }
        hc1->SetBinContent(i,N1/N0);
        hc2->SetBinContent(i,N2/N0);
    }
    hc1->Draw();
    hc2->Draw("same");
}

```

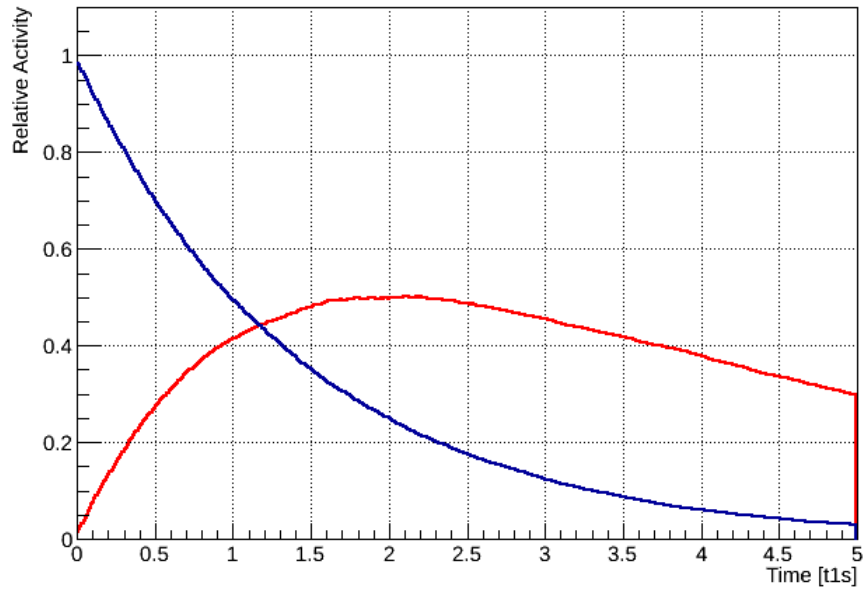
$$t_2 = 10 t_1$$



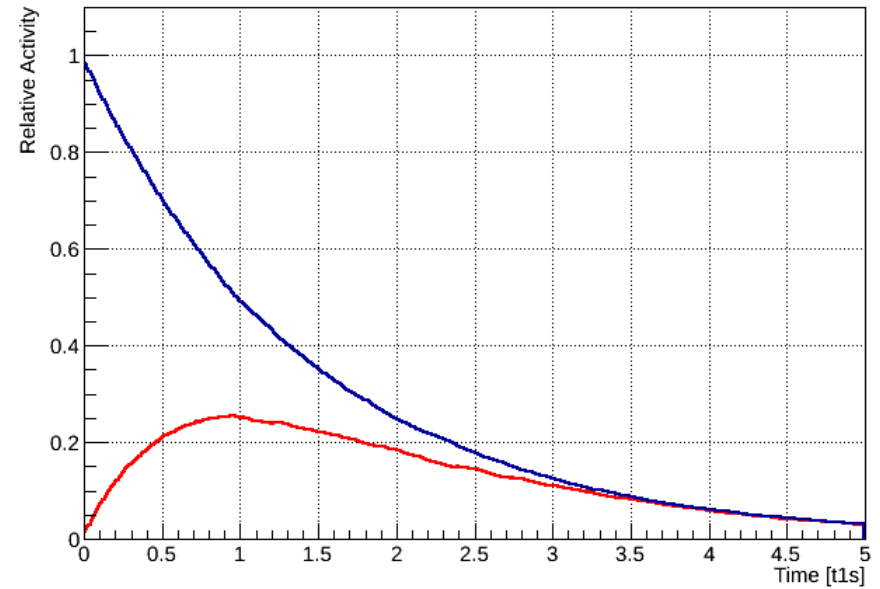
$$t_1 = 10 t_2$$



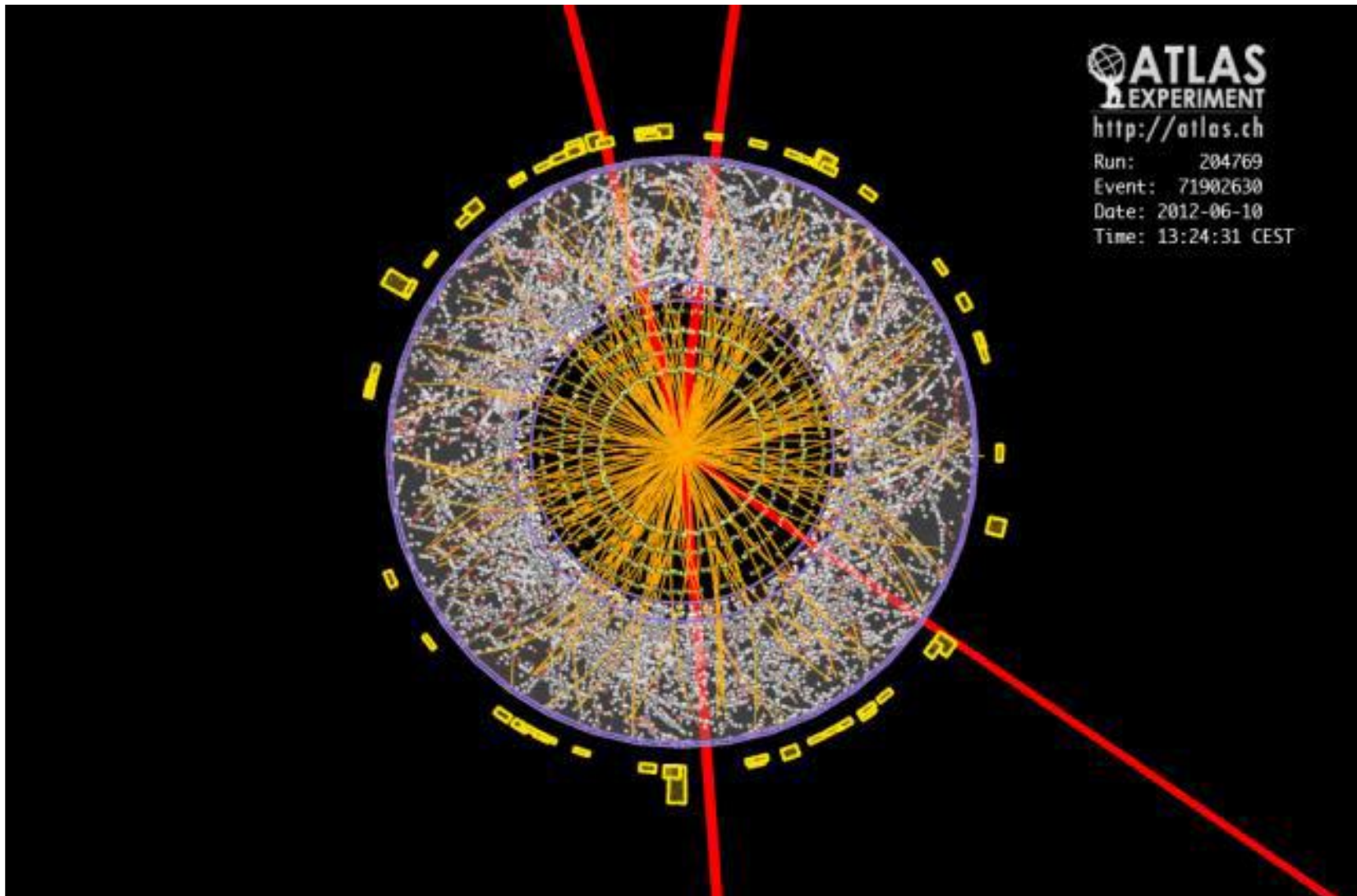
$$t_2 = 2 t_1$$



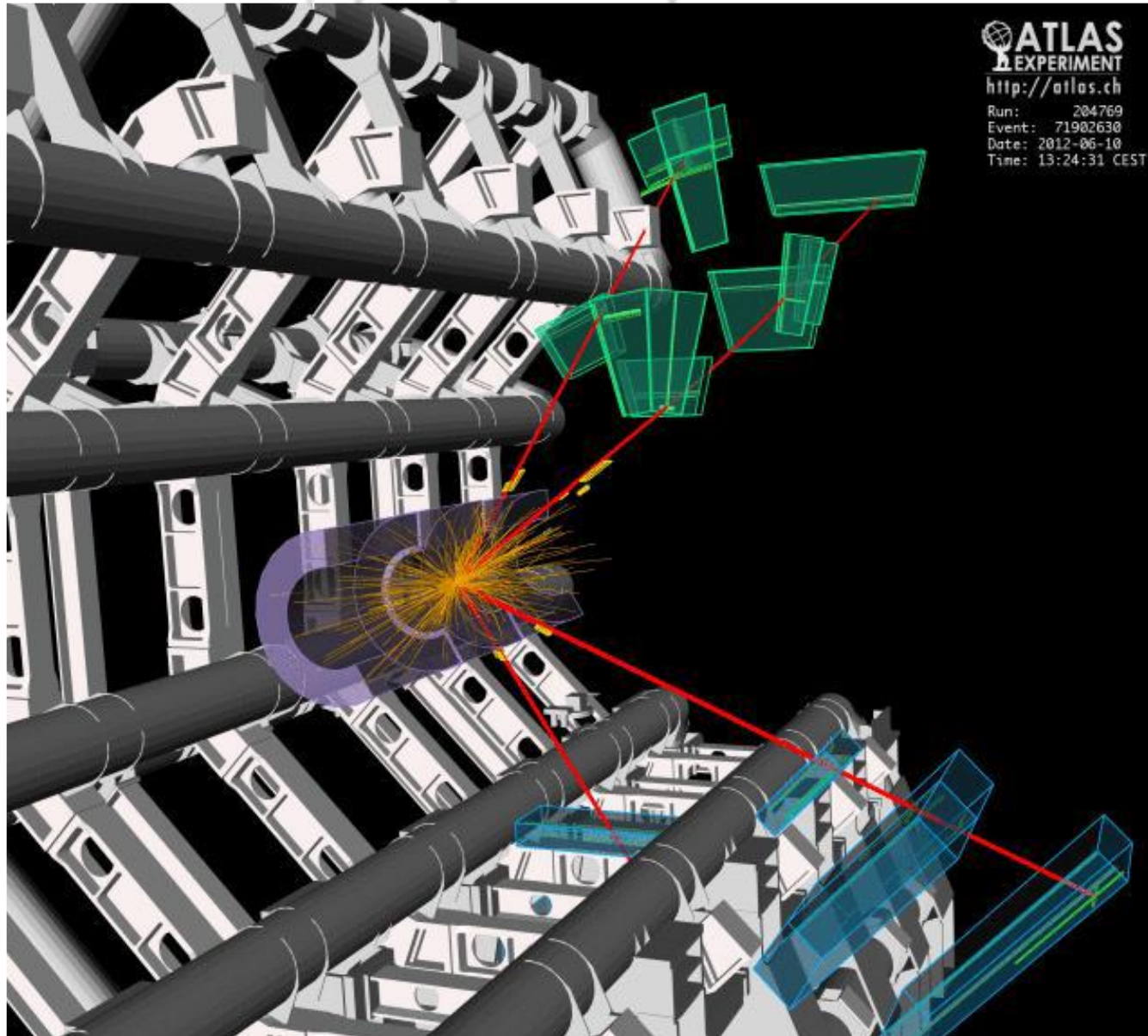
$$t_1 = 2 t_2$$



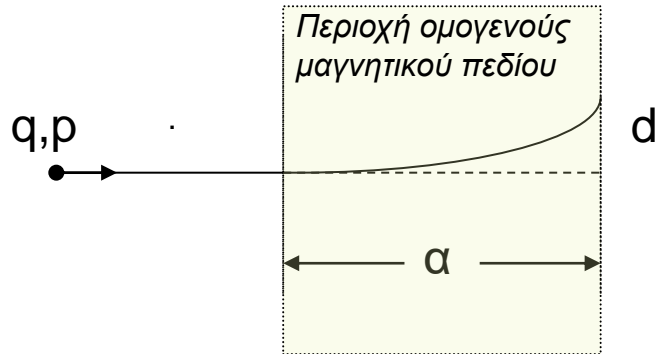
# ΥΠΟΛΟΓΙΣΤΙΚΗ ΦΥΣΙΚΗ - MC Προσομοίωση: Ανακατασκευή φορτισμένων σωματιδίων



φορτισμένων σωματιδίων



# MC Προσομοίωση: Ανακατασκευή φορτισμένων σωματιδίων



Δεδομένων των  $q$ ,  $B$ ,  $a$  και  $d$ , να υπολογιστεί η ορμή  $p$  του σωματιδίου.

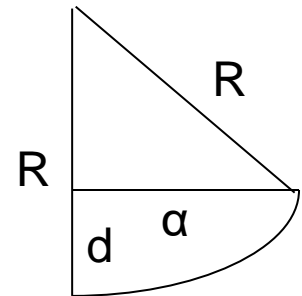
Υπό την επίδραση της δύναμης Lorentz το σωματίδιο διαγράφει κυκλική τροχιά ακτίνας  $R$ , όπου:

$$\mathbf{F} = q(\mathbf{v} \times \mathbf{B})$$

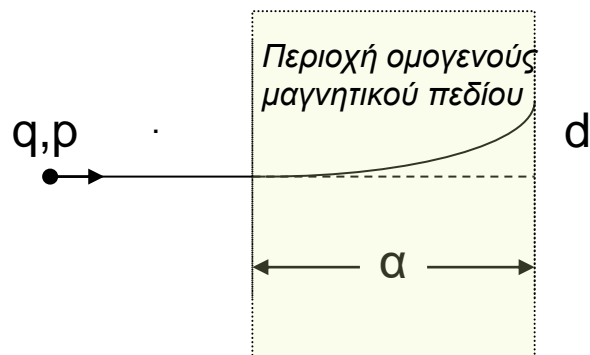
$$qvB = m \frac{v^2}{R} \Rightarrow p = qBR$$

αλλά

$$(R - d)^2 + a^2 = R^2 \Rightarrow R = \frac{a^2 + d^2}{2d}$$



$$p = qB \frac{a^2 + d^2}{2d}$$

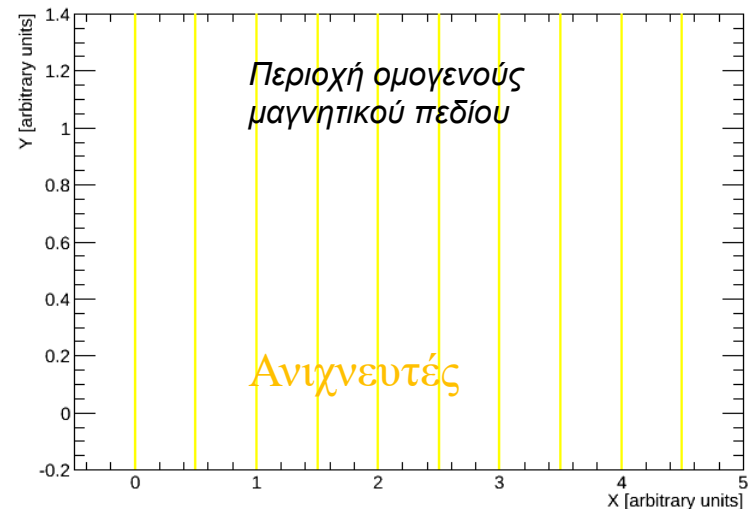
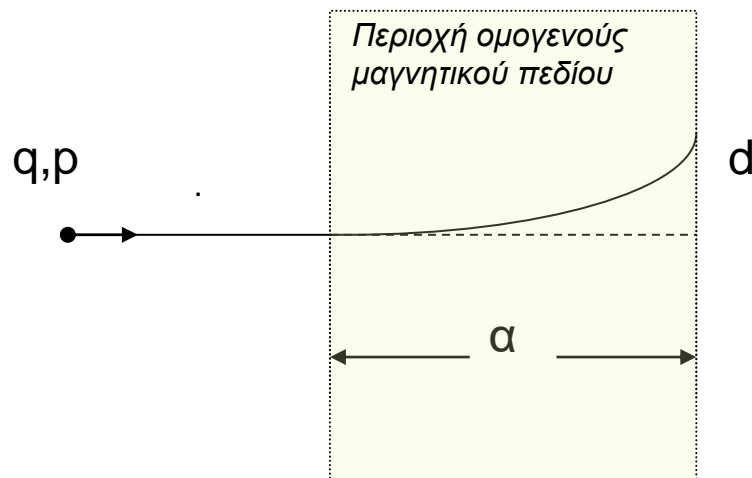


## Απλουστεύσεις στην προσομοίωση:

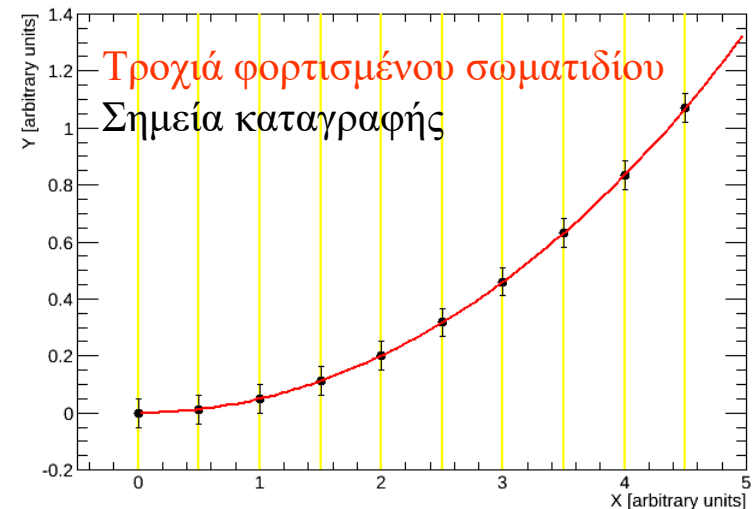
- Επίπεδη και όχι 3-διάστατη κίνηση
- Ομογενές, κάθετο στο επίπεδο μαγνητικό πεδίο
- Θεωρούμε γνωστό το σημείο εισόδου του σωματιδίου
- Αμελούμε σκεδάσεις του φορτισμένου σωματιδίου στο υλικό των ανιχνευτών
- Αμελούμε απώλεια ενέργειας του φορτισμένου σωματιδίου
- Χρησιμοποιούμε μονάδες ώστε  $qB=1 \rightarrow p=R$



# Προσομοίωση: Ανακατασκευή φορτισμένων σωματιδίων



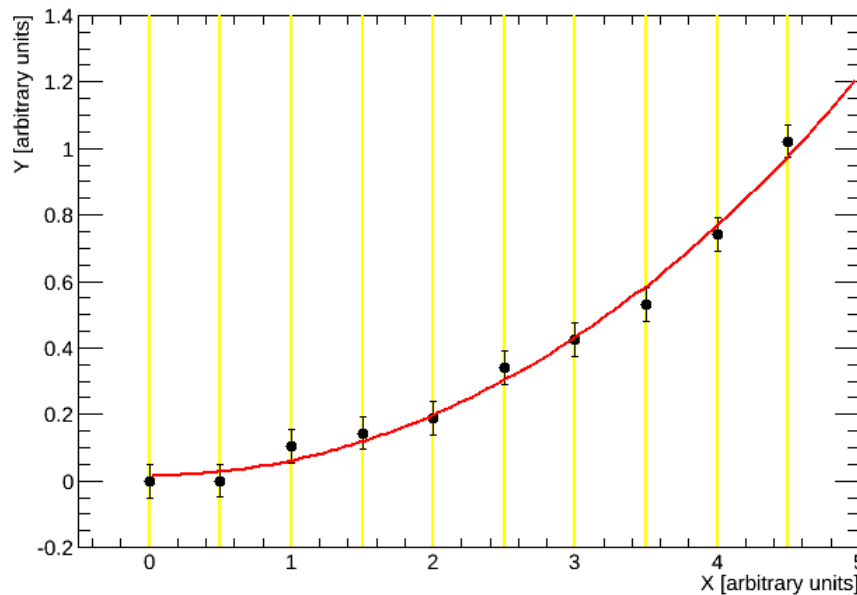
```
double R=10.;  
double x[10], y[10], ey[10], ex[10];  
for(int i=0; i<10;i++)  
{  
    x[i]=i*0.5;  
    y[i]=R-sqrt(R*R-x[i]*x[i]);  
    ey[i]=0.05;  
    ex[i]=0.001;  
}
```





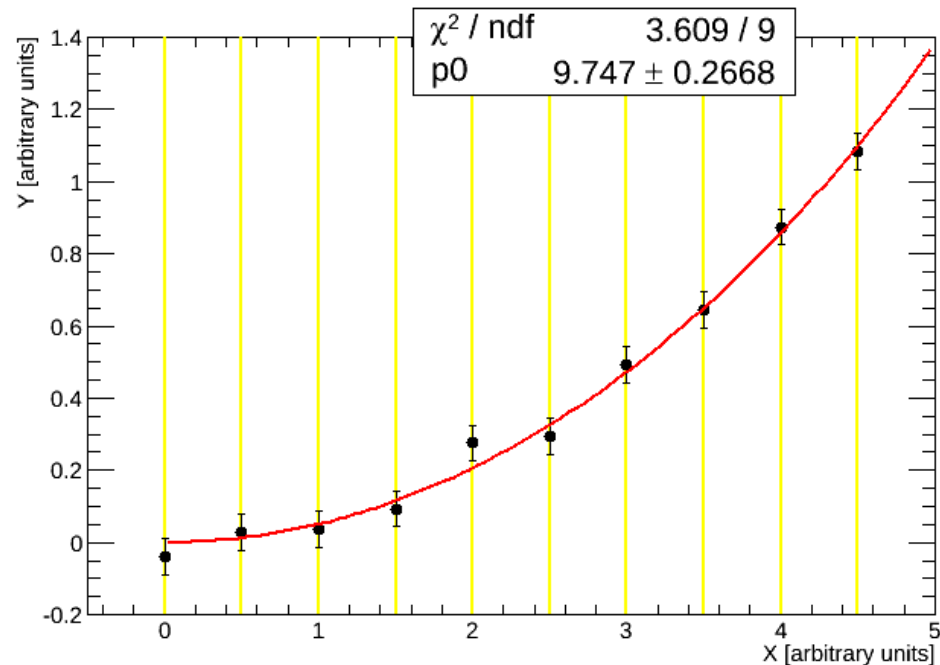
# MC Προσομοίωση: Ανακατασκευή φορτισμένων σωματιδίων

```
double x[10], y[10], ey[10], ex[10];  
for(int i=0; i<10;i++)  
{  
    x[i]=i*0.5;  
    y[i]=R-sqrt(R*R-x[i]*x[i]);  
    ey[i]=0.05;  
    ex[i]=0.001;  
    y[i]=y[i]+ran->Gaus(0.,1.)*ey[i];  
}
```



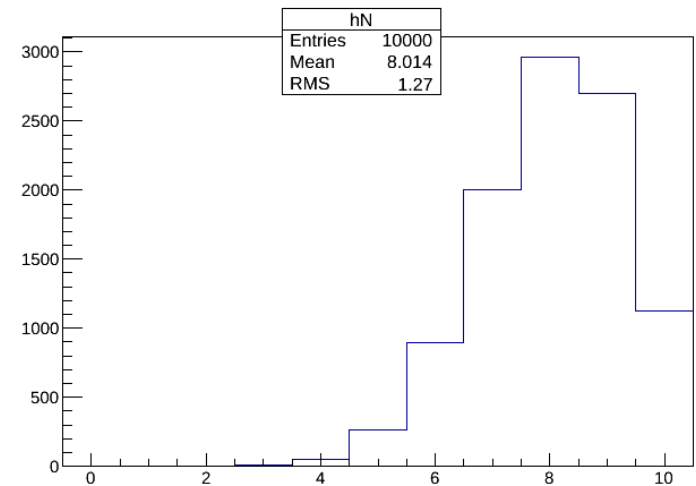
# MC Προσομοίωση: Ανακατασκευή φορτισμένων σωματιδίων

```
TGraphErrors *g=new TGraphErrors(N,xe,ye,ex,ey);  
g->Draw("P");  
TF1 *func = new TF1("func","[0]-sqrt([0]*[0]-x*x)",0.,5.);  
func->SetParameter(0,R);  
g->Fit("func","q","q",0.,5.);
```

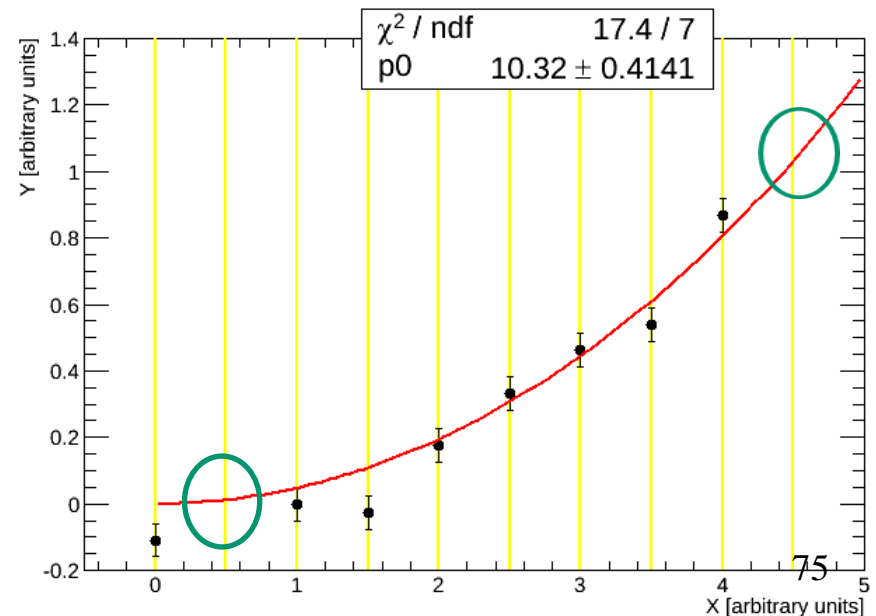


# MC Προσομοίωση: Ανακατασκευή φορτισμένων σωματιδίων

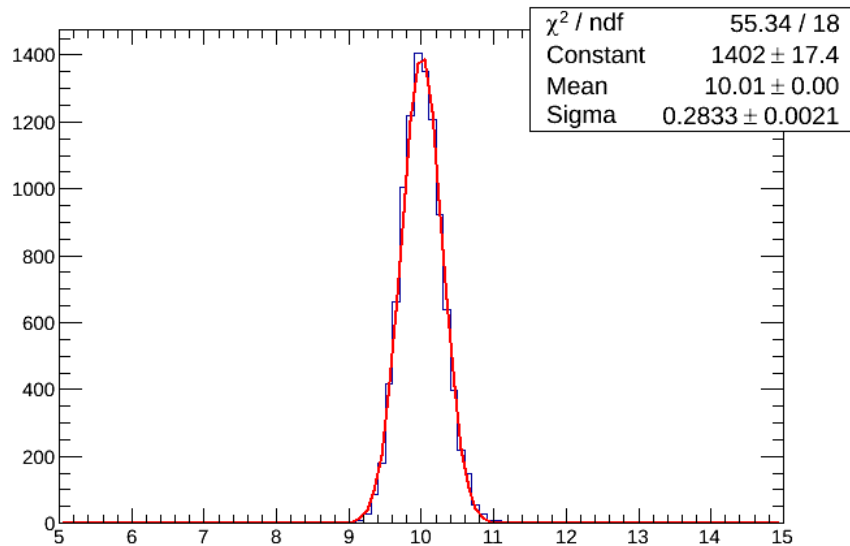
```
double x[10], y[10], ey[10], ex[10];
for(int i=0; i<10;i++)
{
    x[i]=i*0.5;
    y[i]=R-sqrt(R*R-x[i]*x[i]);
    ey[i]=0.05;
    ex[i]=0.001;
    y[i]=y[i]+ran->Gaus(0.,1.)*ey[i];
    id[i]=1;
    if(ran->Rndm()>0.8)id[i]=-1;
}
```



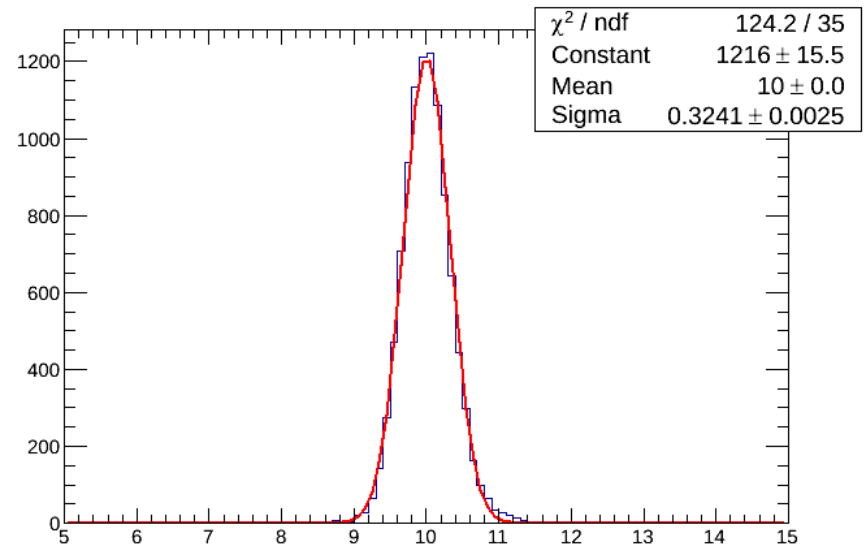
```
int N=0;
for(int i=0; i<10; i++)
{
    if(id[i]>0)
    {
        xe[N]=x[i];
        ye[N]=y[i];
        N++;
    }
}
```



Επαναλαμβάνουμε πολλές φορές και βρίσκουμε την κατανομή των εκτιμήσεων



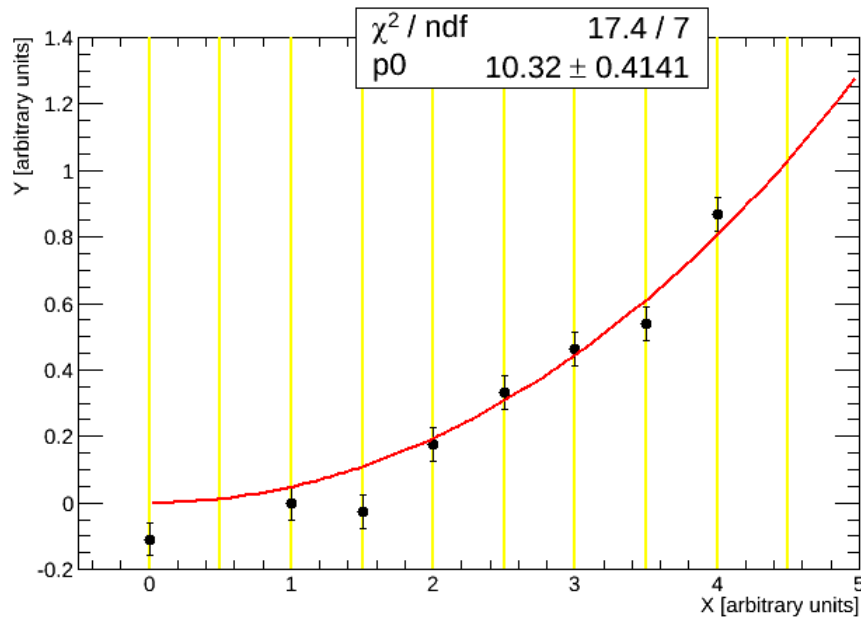
Απόδοση 100%



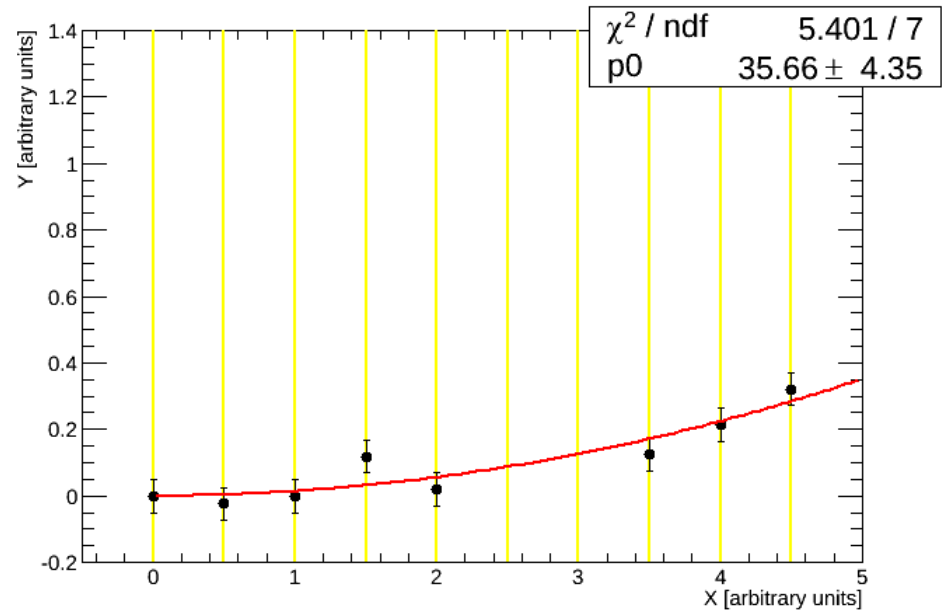
Απόδοση 80%

Όσο αυξάνεται η ορμή η σχετική διακριτική ικανότητα χειροτερεύει

R=10



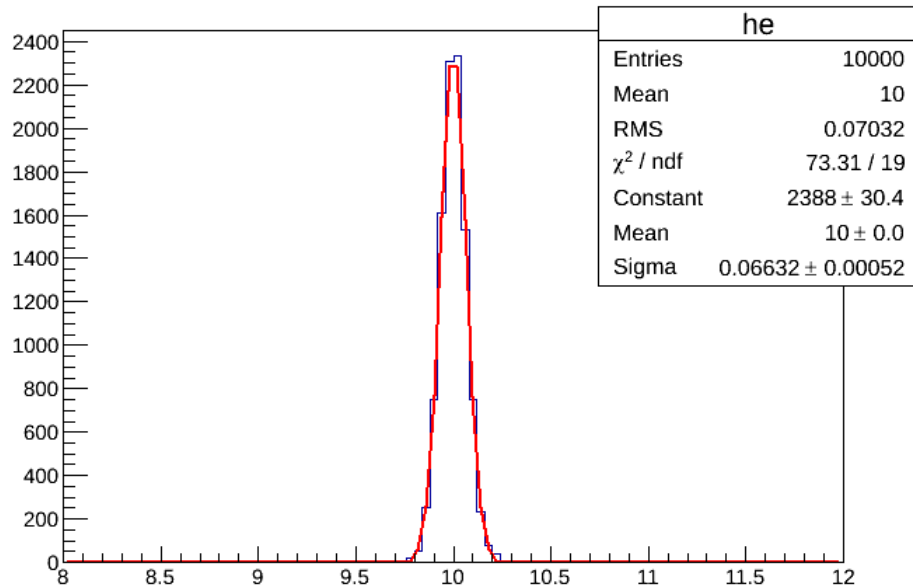
R=40



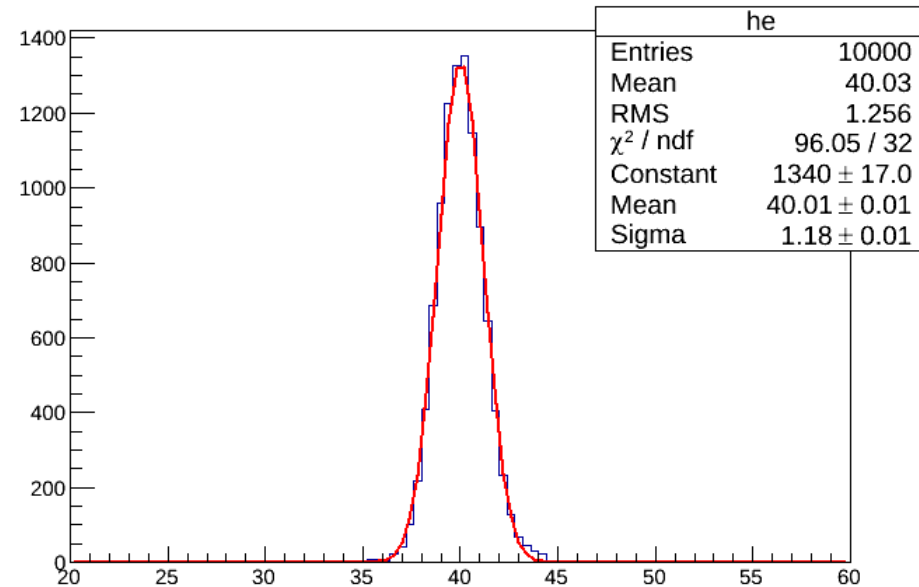
Όσο αυξάνεται η ορμή η σχετική διακριτική ικανότητα χειροτερεύει

$$\frac{dp}{p} \propto p$$

R=10,  $ey=0.01$ ,  $\sigma=0.7\%$



R=40,  $ey=0.01$ ,  $\sigma=2.9\%$



# MC Προσομοίωση: Ανακατασκευή φορτισμένων σωματιδίων

```
void particle_c()
{
    ran=new TRandom3(77331);
    TH1F *he=new TH1F("he"," ",100, 5.,15. );
    TF1 *func = new TF1("func","[0]-sqrt([0]*[0]-x*x)",0.,5.);
    double R=10.; double eff=0.8;
    double x[10],y[10], ey[10], ex[10];
    int id[10]; double xe[10],ye[10];
    for(int itoy=0;itoy<10000;itoy++)
    {
        for(int i=0; i<10;i++)
        {
            x[i]=i*0.5;
            y[i]=R-sqrt(R*R-x[i]*x[i]);
            ey[i]=0.05;
            ex[i]=0.001;
            y[i]=y[i]+ran->Gaus(0.,1.)*ey[i];
            id[i]=1;
            if(ran->Rndm() > eff)id[i]=-1;
        }
        int N=0;
        for(int i=0; i<10; i++)
        {
            if(id[i]>0)
            {
                xe[N]=x[i]; ye[N]=y[i]; N++;
            }
        }
    }
}
```

# MC Προσομοίωση: Ανακατασκευή φορτισμένων σωματιδίων

```
TGraphErrors *g=new TGraphErrors(N,xe,ye,ex,ey);  
//g->Draw("P");  
func->SetParameter(0,R);  
g->Fit("func","q","q",0.,5.);  
he->Fill(func->GetParameter(0));  
}  
he->Draw();  
}
```



# MC: Επίλυση κβαντομηχανικών προβλημάτων

## Η χρονοανεξάρτητη εξίσωση Schrödinger

$$-\nabla^2\psi + V(\vec{r})\psi = E\psi$$

### Εύρεση βασικής κατάστασης



Εύρεση της  $\phi$  που ελαχιστοποιεί την

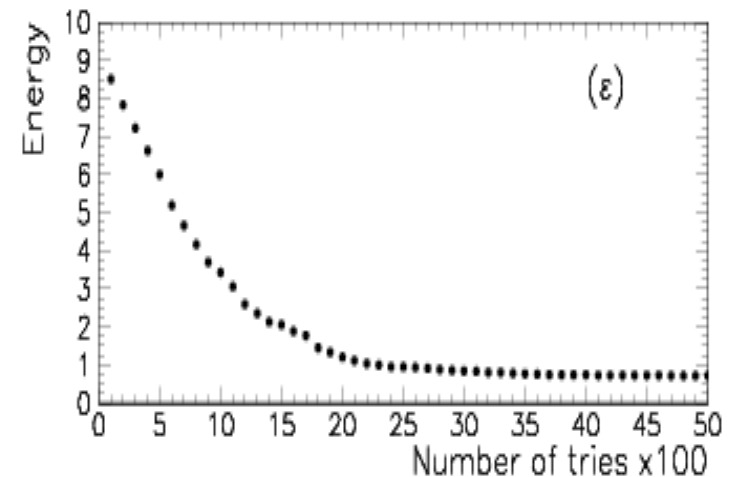
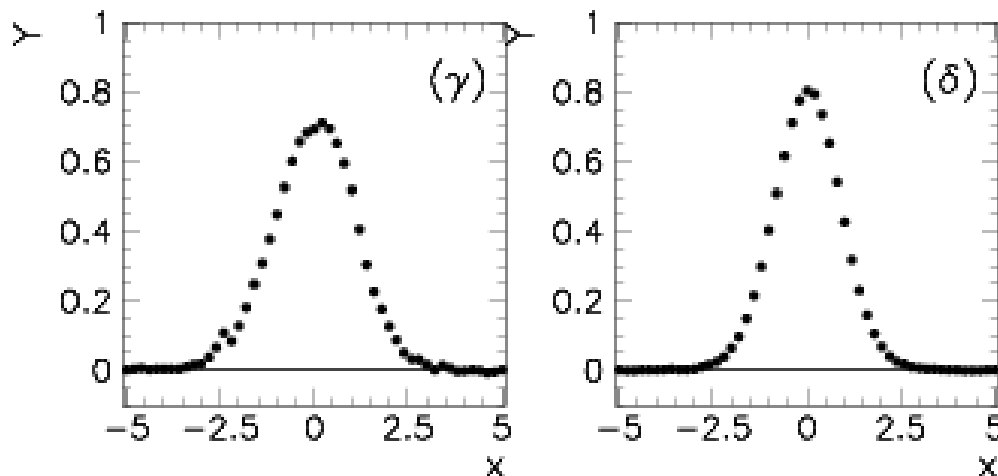
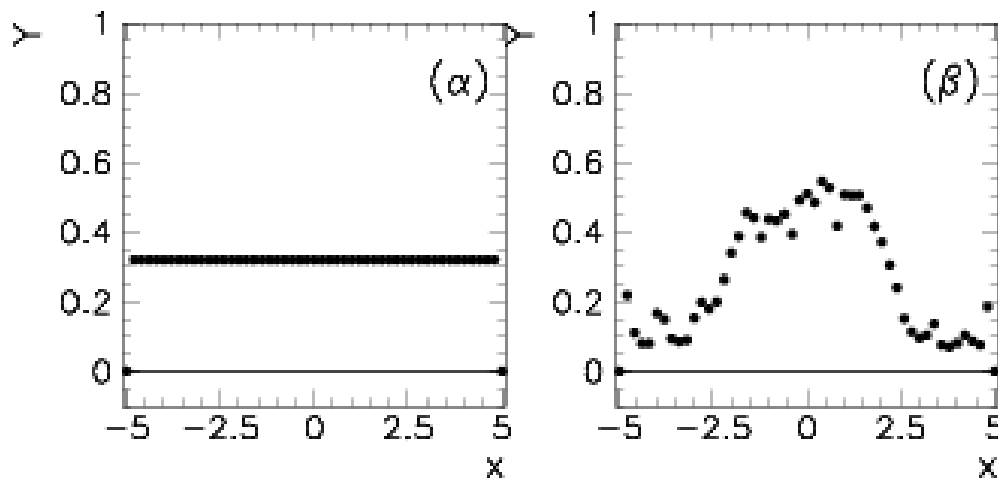
$$E^\phi \equiv \frac{\int \phi^* H \phi d\nu}{\int \phi^* \phi d\nu}$$

### Αλγόριθμος

- Δημιουργούμε ένα χωρικό πλέγμα
- Επιλέγουμε μια αρχική κυματοσυνάρτηση  $\phi$
- Επιλέγουμε τυχαία ένα σημείο του πλέγματος
- Μεταβάλλουμε τη  $\phi$  στο σημείο αυτό κατά ένα ποσό επιλεγμένο τυχαία στην περιοχή  $\pm\delta\phi$
- Υπολογίζουμε την ενέργεια της νέας κατάστασης και αν είναι μικρότερη της προηγούμενης κρατάμε την αλλαγή και έχουμε μια νέα δοκιμαστική συνάρτηση / αν όχι επιστρέφουμε στην προηγούμενη (Αλγόριθμος Metropolis)
- Επαναλαμβάνουμε τα τρία τελευταία βήματα

# MC: Επίλυση κβαντομηχανικών προβλημάτων

## Η χρονοανεξάρτητη εξίσωση Schrödinger Αρμονικός Ταλαντωτής



# Προσομοίωση MC: Μοντέλο Ising

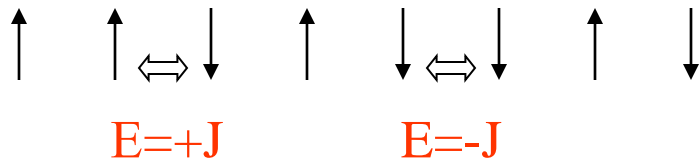
Παράδειγμα για το πώς οι μικροσκοπικές αλληλεπιδράσεις καθορίζουν μακροσκοπικά μεγέθη.

Σιδηρομαγνητισμός : κβαντομηχανικό μοντέλο Ising

Στατιστική φυσική:  $P(a_j) = \frac{e^{-E(a_j)/kT}}{Z(T)}$   $Z(T) = \sum_{a_j} e^{-E(a_j)/kT}$

όσο μεγαλύτερη είναι η ενέργεια μιας κατάστασης τόσο μικρότερη είναι η πιθανότητά της

Μοντέλο: N μαγνητικά δίπολα τοποθετημένα σε ένα κρυσταλλικό πλέγμα



$$|a_j\rangle = |s_1, s_2, \dots, s_N\rangle = \left| \pm \frac{1}{2}, \pm \frac{1}{2}, \dots, \pm \frac{1}{2} \right\rangle \quad j = 1, 2^N$$

# Προσομοίωση MC: Μοντέλο Ising

Δυναμικό για το σπιν i:  $V_i = -J\vec{s}_i \cdot \vec{s}_{i+1} - g\mu\vec{s}_i \cdot \vec{B}$

(αλληλεπίδραση μόνο με τα γειτονικά σπιν)

Ενέργεια του συστήματος :

(μέση τιμή σε όλα τα σπιν )

**Εξαρτάται από πρόσημο του J**

$$\begin{aligned} E(a) &= \left\langle a \left| \sum_i V_i \right| a \right\rangle \\ &= -J \sum_{i=1}^{N-1} s_i s_{i+1} - B\mu \sum_{i=1}^N s_i \end{aligned}$$

Αναλυτική λύση σε **μία** διάσταση:

$$\begin{aligned} \frac{U}{J} &= -N \frac{e^{J/kT} - e^{-J/kT}}{e^{J/kT} + e^{-J/kT}} \\ &= \begin{cases} N, kT \rightarrow 0 \\ 0, kT \rightarrow \infty \end{cases} \end{aligned}$$

$$C(kT) = \frac{1}{N} \frac{dU}{dT} = \frac{(J/kT)^2}{(e^{J/kT} + e^{-J/kT})^2} = \frac{(J/kT)^2}{\cosh^2(J/KT)}$$

$$M(kT) = \frac{Ne^{J/kT} \sinh(B/kT)}{\sqrt{e^{2J/kT} \sinh^2(B/kT) + e^{-2J/kT}}}$$

# Προσομοίωση MC: Μοντέλο Ising

Αριθμητική λύση ισοδύναμη με ολοκλήρωση σε  $2^N$  διαστάσεις

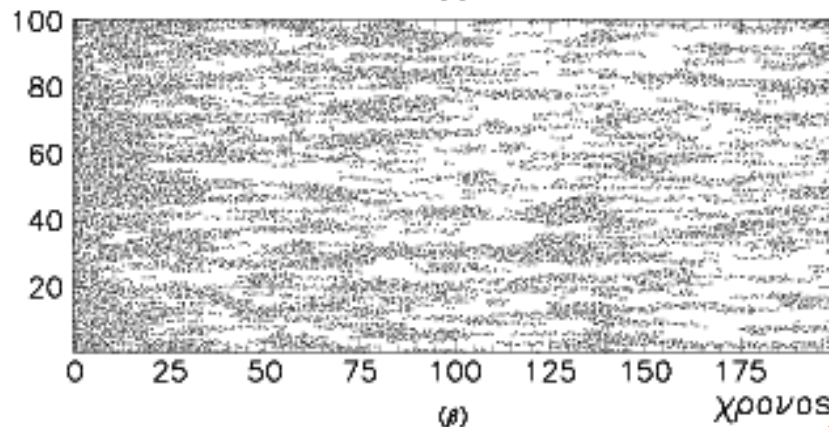
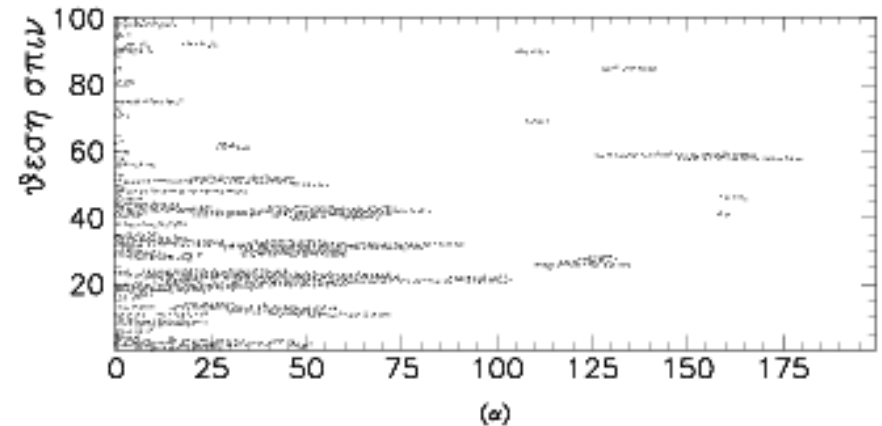
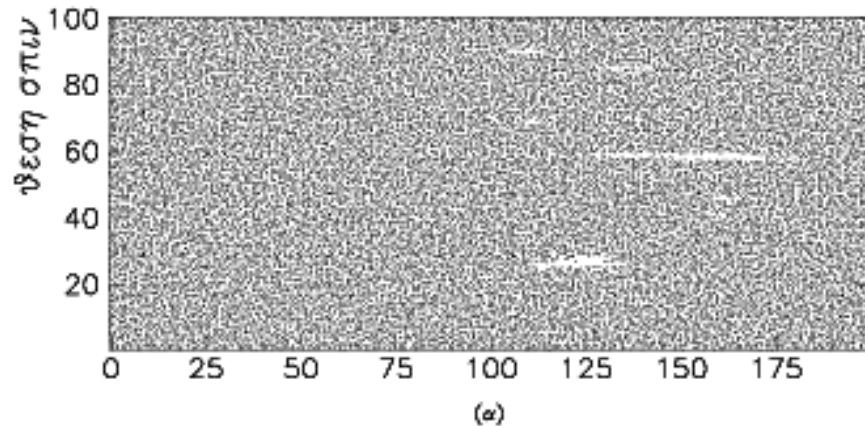
## Αλγόριθμος Metropolis:

- I. Αρχίζουμε με μία αυθαίρετη διάταξη των σπιν (αρχική συνθήκη),  $\alpha$
- II. Για να γεννήσουμε μια νέα δοκιμαστική διάταξη, αδοκ:
  1. Διαλέγουμε τυχαία ένα από τα σωματίδια
  2. Αντιστρέφουμε το σπιν του
  3. Υπολογίζουμε την ενέργεια της νέας (δοκιμαστικής) διάταξης
  4. Αν  $E(\alpha_{δοκ}) < E(\alpha)$ , δεχόμαστε την αλλαγή
  5. **Αν  $E(\alpha_{δοκ}) > E(\alpha)$** 
    - Υπολογίζουμε το  $P = \exp(-\Delta E/kT)$
    - Γεννάμε ομοιόμορφα τυχαίο αριθμό  $r[0,1]$
    - Αν  $P \geq r$ , δεχόμαστε τη δοκιμαστική διάταξη
    - Αν  $P < r$ , απορρίπτουμε τη δοκιμαστική διάταξη
- III. Επαναλαμβάνουμε το II.
- IV. Περιοδικές συνοριακές συνθήκες.

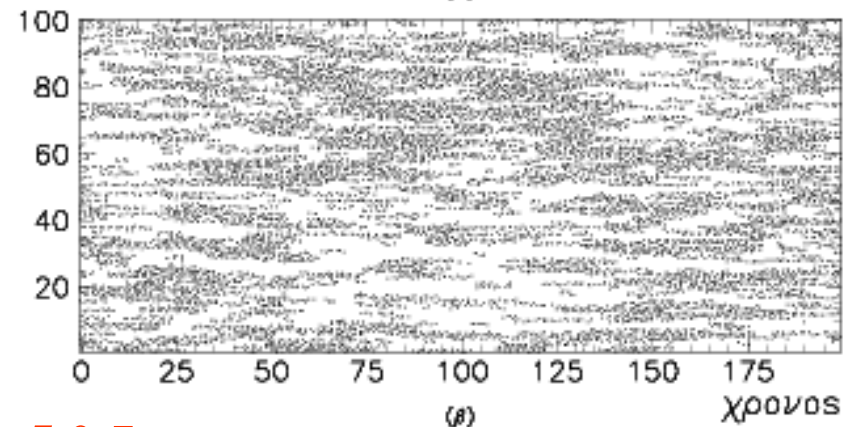
# Προσομοίωση MC: Μοντέλο Ising

Εφαρμογή 2-d (πλέγμα 10x10)

$kT=1.5 \text{ J}$



$kT=5.0 \text{ J}$

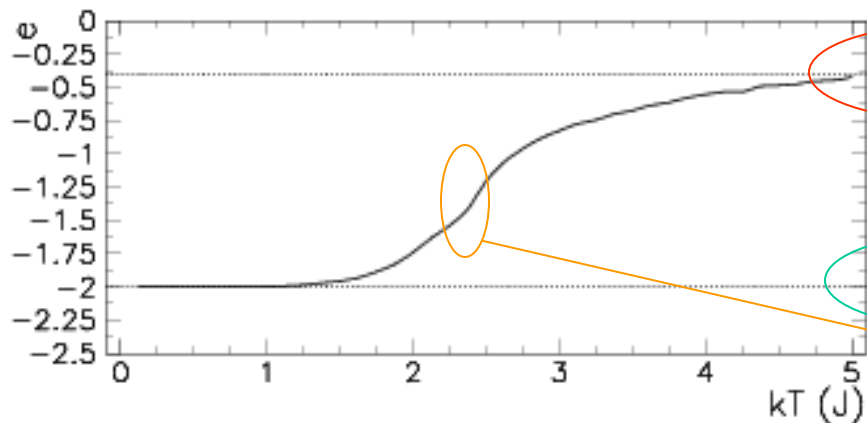


«Ψυχρό» ξεκίνημα

«Θερμό» ξεκίνημα

# Προσομοίωση MC: Μοντέλο Ising

Ενέργεια



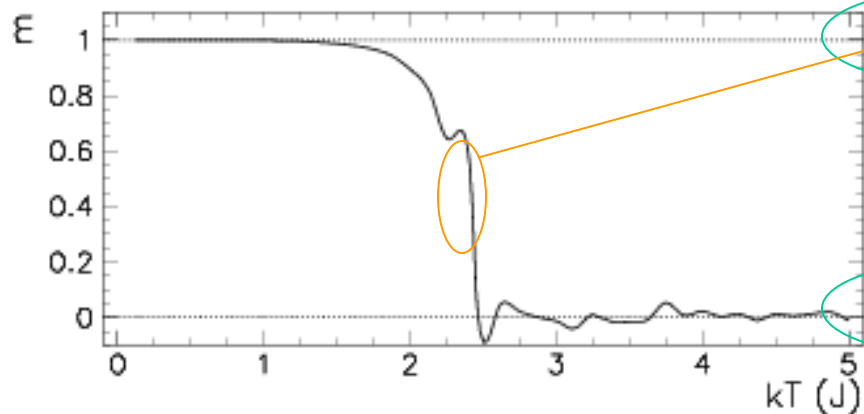
-0.4J

-2.J

Αλλαγή φάσης

$|m|=1$

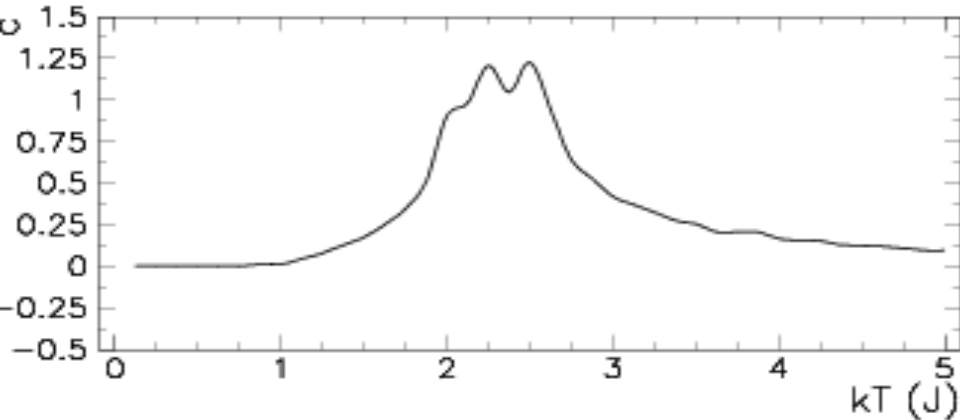
$m=0$



Μαγνήτιση

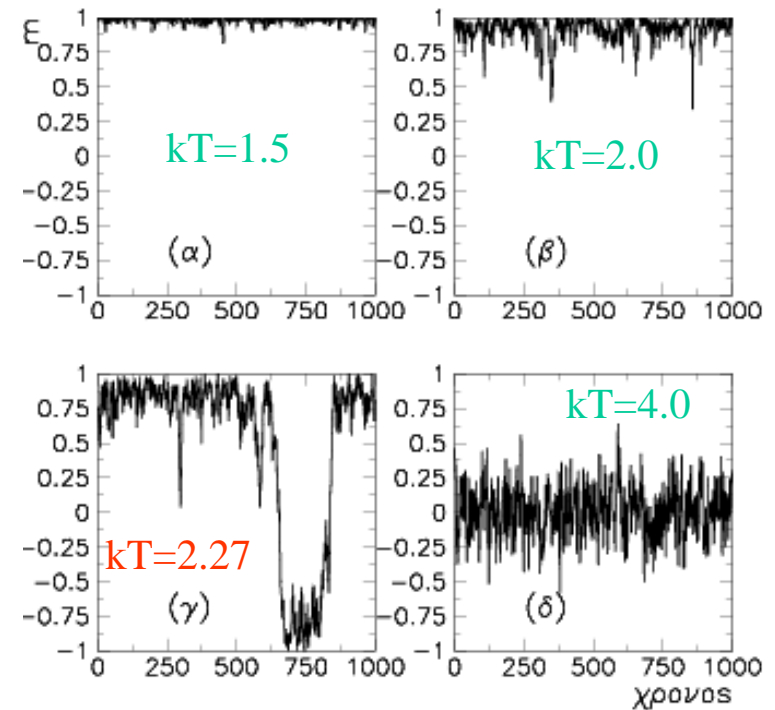
# Προσομοίωση MC: Μοντέλο Ising

Ειδική θερμότητα



$$C = \frac{(\Delta E)^2}{kT^2}$$

Αλλαγή φάσης για  $kT=2.27$  J



Μελέτη διακυμάνσεων



# Προσομοίωση MC: Διάδοση νετρονίων

- Θεωρούμε άπειρο επίπεδο σε x,y με πάχος t σε z
- Υπάρχει πιθανότητα :  $p_c$  να συλληφθεί  
και :  $p_s$  να σκεδαστεί
- Αν σκεδαστεί θεωρούμε ομοιόμορφη γωνιακή κατανομή  
 $p(\theta, \varphi) d\theta d\varphi = d\Omega/4\pi$
- Δεδομένου ότι επίπεδο άπειρο σε x, y, δεν ενδιαφερόμαστε για γωνία  $\varphi$ .

$$d\Omega = \sin\theta d\theta d\varphi \quad \Rightarrow \quad p(\theta, \varphi) = \frac{\sin\theta}{4\pi}$$

$$p(\theta) = \int_0^{2\pi} p(\theta, \varphi) d\varphi = \frac{1}{2} \sin\theta \quad \Rightarrow \quad \cos\theta = 1 - 2R$$

$$p(\varphi) = \int_0^\pi p(\theta, \varphi) d\theta = \frac{1}{2\pi} \quad \Rightarrow \quad \varphi = 2\pi R$$

$$R = \int_0^\theta \frac{1}{2} \sin x dx$$

- Το μέσο μήκος μεταξύ δύο σκεδάσεων δίνεται από την πιθανότητα:

$$p(l) \propto e^{-l/\lambda} \quad \Rightarrow \quad l = -\lambda \ln R$$

# Προσομοίωση MC: Διάδοση νετρονίων

## Αλγόριθμος

1. Απόφαση αν το νετρόνιο συλλαμβάνεται ή σκεδάζεται. Αν συλλαμβάνεται αυξάνει κατά ένα ο αριθμός  $N_\sigma \rightarrow$  βήμα 4.
2. Αν το νετρόνιο σκεδάζεται, υπολογισμός  $\theta$  και  $l$ . Αντίστοιχη αλλαγή του  $z$
3. Αν  $z < 0$ , αυξάνει κατά ένα ο αριθμός  $N_o >>$  βήμα 4.  
Αν  $z > t$ , αυξάνει κατά ένα ο αριθμός  $N_\delta >>$  βήμα 4.  
Αν τίποτα από τα δύο  $\rightarrow$  βήμα 1
4. Επανάληψη των βημάτων 1-3 για κάθε νέο νετρόνιο.

# MC: Προσομοίωση φασικού χώρου

Σκέδαση δύο  
σωματιδίων

$$d\sigma = \frac{|M|^2}{F} dQ$$

Στοιχείο  
πίνακα

Παράγοντας  
Φασικού χώρου

$$dQ = (2\pi)^4 \delta^4(p_c + p_d - p_a - p_b) \frac{d^3 p_c}{(2\pi)^3 2E_c} \frac{d^3 p_d}{(2\pi)^3 2E_d}$$

Διάσπαση β

$$d\Gamma = \frac{|M|^2}{2E_A} dQ$$

Διατήρηση  
τετραορμής

$$dQ = (2\pi)^4 \delta^4(p_A + p_1 - p_2 - p_3) \frac{d^3 p_1}{(2\pi)^3 2E_1} \frac{d^3 p_2}{(2\pi)^3 2E_2} \frac{d^3 p_3}{(2\pi)^3 2E_3}$$

# MC: Προσομοίωση φασικού χώρου

## Υπολογισμός μια παραμέτρου του υπό μελέτη συστήματος:

Πως από τα μετρήσιμα μεγέθη, π.χ. τις ορμές των σωματιδίων που μπορούν να μετρηθούν θα μπορέσουμε να εκτιμήσουμε την άγνωστη παράμετρο;

- Προσομοίωση του φασικού χώρου
- Προσομοίωση της απόκρισης της μετρητικής διάταξης
- Σύγκριση αποτελεσμάτων της πραγματικής μέτρησης με αυτά της προσομοίωσης (Κατάλληλη στατιστική επεξεργασία των δεδομένων οδηγεί σε εκτίμηση της παραμέτρου)

Η διαδικασία της ολοκλήρωσης του φασικού χώρου συνδέεται άμεσα με τη διαδικασία προσομοίωσης.

Μπορούμε να μιλάμε για μεθοδολογίες «γεννητόρων» για την επίλυση των παραπάνω προβλημάτων από κοινού, με την έννοια ότι η τεχνική Monte Carlo χρησιμοποιείται σε αμφότερες τις περιπτώσεις για τη δημιουργία («γέννηση») δείγματος γεγονότων.

# MC: Προσομοίωση φασικού χώρου

Στη γενική περίπτωση  
το ολοκλήρωμα του φασικού χώρου  $n$  σωματιδίων

$$R_n = \int_{4n} \delta^4(P - \sum_{j=1}^n p_j) \prod_{i=1}^n \delta(p_i^2 - m_i^2) d^4 p_i$$

Διατήρηση  
ενέργειας  
και ορμής

Η ενέργεια κάθε σωματιδίου  
πρέπει να είναι συμβατή με  
την ορμή του

Εκτός από τη διατήρηση της ορμής και της ενέργειας  
θα πρέπει να εκφράζεται σωστά και η πυκνότητα των  
καταστάσεων. Αναμένουμε μεγαλύτερη πιθανότητα να  
βρούμε ένα γεγονός σε μια περιοχή με περισσότερες  
δυνατές καταστάσεις

$$\delta(p_i^2 - m_i^2) d^4 p_i = \frac{p_i^2}{E_i} dp_i d\cos\theta_i d\varphi_i$$

# MC: Προσομοίωση φασικού χώρου

## Γεννήτορες φασικού χώρου τύπου T (κινητικής ενέργειας)

$$R_n = \int \frac{d^3 p_1 d^3 p_2 \cdots d^3 p_n \delta^3(\sum p_i) \delta(E_0 - \sum E_i)}{E_1 E_2 \cdots E_n}$$

- Αναπαραστατικό
- Υπολογίζουμε τα επιτρεπτά κινηματικά όρια των συνιστωσών των ορμών και τις επιλέγουμε όλες τυχαία μεταξύ των ορίων αυτών.
  - Εκτός από τις τελευταίες τέσσερις, οι οποίες υπολογίζονται από την ολοκλήρωση των συναρτήσεων δ. Δηλαδή, ελέγχουμε αν υπάρχουν τέσσερις συνιστώσες που να ικανοποιούν τις εξισώσεις διατήρησης της ορμής και της ενέργειας.
  - Αν δεν υπάρχουν (αυτή θα είναι και η συνήθης περίπτωση), τότε επιλέγουμε ένα καινούργιο δείγμα από  $3n-4$  τυχαίες συνιστώσες ορμών.
  - Αν βρούμε ένα δείγμα που ανήκει στην φυσικά επιτρεπόμενη περιοχή, τότε το αθροίζουμε δίνοντάς του ένα βάρος ίσο με το αντίστροφο του γινομένου των ενεργειών των σωματιδίων

$$R_n(P; m_1, m_2, \dots, m_n) = \int R_{n-1}(P - p_n; m_1, m_2, \dots, m_{n-1}) \frac{d^3 p_n}{2E_n}$$

## MC: Προσομοίωση φασικού χώρου

$$R_n(P; m_1, m_2, \dots, m_n) = \int R_{n-1}(P - p_n; m_1, m_2, \dots, m_{n-1}) \frac{d^3 p_n}{2E_n}$$



### Γεννήτορες φασικού χώρου τύπου M (Αναλλοίωτης μάζας)

$$R_n(P; m_1, m_2, \dots, m_n) = \int R_{n-l+1}(P; M_l, m_{l+1}, \dots, m_n) R_l(P_l; m_1, m_2, \dots, m_l) dM_l^2$$

### Για δύο σωματίδια:

$$R_2(M_{i+1}; m_i, m_{i+1}) = \frac{2\pi}{M_{i+1}} \sqrt{M_{i+1}^2 + \left(\frac{M_i^2 - m_{i+1}^2}{M_{i+1}}\right)^2 - 2(M_i^2 + m_{i+1}^2)}$$

$$R_n = \frac{1}{m_1} \int \cdots \int \prod_{i=1}^{n-1} [2M_i R_2(M_{i+1}; M_i, m_{i+1})] dM_{n-1} \cdots dM_2$$

### Δύο προβλήματα:

α) Θα έπρεπε να έχουμε  $3n-4$  μεταβλητές για ολοκλήρωση και έχουμε μόνο  $n-2$ .

Τι έγιναν οι υπόλοιπες;

β) Ποια είναι τα όρια στις ολοκληρώσεις των αναλλοίωτων μαζών;

Πως μπορούμε να τα επιλέξουμε αποδοτικά;

## MC: Προσομοίωση φασικού χώρου

(α) Σε κάθε κορυφή, η οποία περιλαμβάνει δύο «σωματίδια» με μάζες  $M_i$  και  $m_{i+1}$ , η παραπάνω έκφραση μπορεί να διατηρήσει την ενέργεια, αλλά τίποτα άλλο. Παραμένουν λοιπόν δύο γωνιακές μεταβλητές για να συμπληρώσουν την εικόνα, οι οποίες όμως θα πρέπει να οριστούν στο αδρανειακό σύστημα αναφοράς του κέντρου μάζας των σωματιδίων αυτών, αφού μόνο στο σύστημα αυτό οι ορμές των προϊόντων της διάσπασης είναι αντίθετες στο χώρο. Έτσι λοιπόν διαλέγουμε ισοτροπικά  $\cos\theta$  και  $\varphi$  στο σύστημα του κέντρου μάζας του ζεύγους των σωματιδίων και στη συνέχεια μετατρέπουμε κατά Lorentz τις ορμές στο εργαστηριακό σύστημα αναφοράς.

(β) Τα φυσικά όρια ολοκλήρωσης ως προς τη μάζα είναι τα όρια που επιτρέπουν κάθε κορυφή να είναι «εξώθερμη». Δηλαδή, για κάθε  $M_i$  θα έχουμε:

$$M_{i-1} + m_i < M_i < M_{i+1} - m_{i+1}$$

Φαίνεται ότι αντιμετωπίζουμε το ίδιο πρόβλημα που αντιμετωπίσαμε και στους γεννήτορες τύπου T, αφού τα όρια του κάθε ολοκληρώματος δεν είναι σταθερά, αλλά εξαρτώνται από τα προηγούμενα ολοκληρώματα.

Παρά ταύτα, στην περίπτωση αυτή το πρόβλημα που αντιμετωπίζουμε είναι ανάλογο του προβλήματος της ολοκλήρωσης με τριγωνικό όριο, οπότε θα έχει και ανάλογη λύση. Η λύση αυτή αντιστοιχεί στη μεθοδολογία της **αναδίπλωσης**.

$$0 < \mathbf{M}_1 < \dots < \mathbf{M}_i < \mathbf{M}_{i+1} < \dots < \mathbf{M}_{n-2} < \mathbf{M}_{\text{tot}}$$



# MC: Προσομοίωση φασικού χώρου

## Παράδειγμα: Διάσπαση του μιονίου

$$d\Gamma = \frac{1}{(2\pi)^5} \frac{1}{16M_\mu} |M|^2 dE_1 dE_2 d\varphi_1 d\cos\theta_1 d\varphi_2$$

Αν λάβουμε μέσο όρο στις καταστάσεις του σπιν:

$$\begin{aligned} d\Gamma &= \frac{1}{(2\pi)^3} \frac{1}{8M_{tot}} |M|^2 dE_1 dE_2 \\ &= \frac{1}{(2\pi)^3} \frac{1}{32M_{tot}^3} |M|^2 dm_{12}^2 dm_{23}^2 \end{aligned}$$

Με τη λογική των γεννητόρων τύπου M:

$$d\Gamma = \frac{1}{(2\pi)^5} \frac{1}{16M_{tot}^2} |M|^2 |\vec{p}_1^*| |\vec{p}_3| dm_{12} d\Omega_1^* d\Omega_3$$

$$|\vec{p}_1^*| = \frac{\left[ \left( m_{12}^2 - (m_1 + m_2)^2 \right) \left( m_{12}^2 - (m_1 - m_2)^2 \right) \right]^{1/2}}{2m_{12}}$$

$$|\vec{p}_3| = \frac{\left[ \left( M_{tot}^2 - (m_{12} + m_3)^2 \right) \left( M_{tot}^2 - (m_{12} - m_3)^2 \right) \right]^{1/2}}{2M_{tot}}$$

# MC: Προσομοίωση φασικού χώρου

```
SUBROUTINE FASIKOS_XOROS13 (AMAS1,AMAS2,AMAS3,AMAS4,N)
```

```
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
REAL AMAS1,AMAS2,AMAS3,AMAS4,RNDM,AMAS12
```

```
DOUBLE PRECISION P,E,TEMP,S,S2,PI, AG,EG,TG
```

```
COMMON/PS/P(6,4)
```

```
INTEGER N,I,J
```

```
EXTERNAL ME2
```

```
DOUBLE PRECISION ME2
```

```
* Initial Particle at rest
```

```
P(1,1)=0.
```

```
P(2,1)=0.
```

```
P(3,1)=0.
```

```
P(5,1)=AMAS1
```

```
P(6,1)=DSQRT(P(1,1)**2+P(2,1)**2+P(3,1)**2)
```

```
P(4,1)=DSQRT(P(5,1)**2+P(6,1)**2)
```

```
* Masses of decay products
```

```
P(5,2)=AMAS2
```

```
P(5,3)=AMAS3
```

```
P(5,4)=AMAS4
```

```
* Initialization
```

```
PI=DACOS(-1.D0)
```

```
S=0.
```

```
S2=0.
```

```
* Loops to perform integration
```

```
DO 100 I=1,N
```

```
* Random choice of 1-2 mass system
```

```
AMAS12=RNDM(DUMMY)*P(5,1)
```

```
* Define electron momentum at 1-2 rest frame
```

```
P(6,2)=(1.D0/2.D0/AMAS12)*
```

```
& DSQRT((AMAS12**2-(P(5,2)+P(5,3))**2)*
```

```
& (AMAS12**2-(P(5,2)-P(5,3))**2))
```

# MC: Προσομοίωση φασικού χώρου

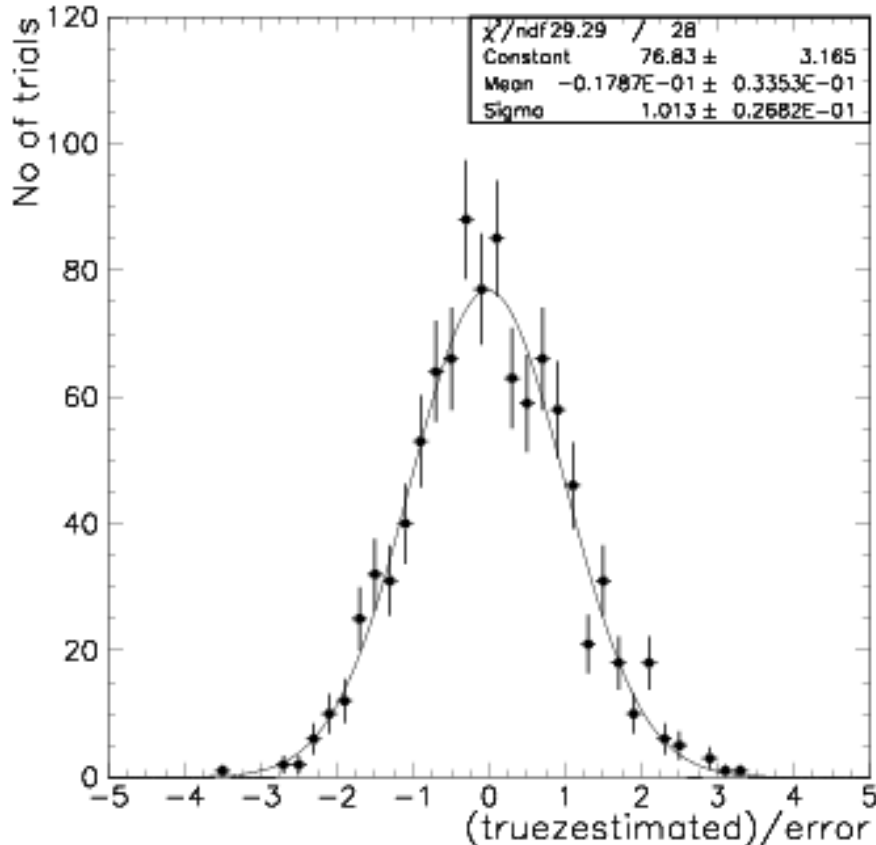
```
* Define neutrino electron momentum
  P(6,4)=(1.D0/2.D0/P(5,1))*
  & DSQRT((P(5,1)**2-(AMAS12+P(5,4))**2)
  & *(P(5,1)**2-(AMAS12-P(5,4))**2))
  TEMP = ME2(DBLE(AMAS1),P(6,4))
  & * P(6,2) * P(6,4)
  & / (2*pi)**5
  & / (16.*P(5,1)**2)
  S = S+TEMP
  S2= S2+TEMP**2
100  CONTINUE
* Volume P(5,1) for random amas12 * 4pi (Omega1) * 4pi (Omega2)
  VOLUME=P(5,1)*(4*pi)**2
* Arithmetic estimation of Width
  AG(1)=VOLUME*S/N
* Error on arithmetic estimation
  EG(1)=VOLUME*DSQRT((S2/N-(S/N)**2)/N)
* Analytic estimation of Width in units of Gfermi**2
  TG(1)=AMAS1**5/192./PI**3
RETURN
END
```

## **FUNCTION ME2(MMU, ENUE)**

```
IMPLICIT NONE
DOUBLE PRECISION ME2,MMU,ENUE
* Matrix element square
* mu--> e nue numu case = 64G**2(p3p2)(p4p1)
* = 32G**2(m**2-2m p4)mp4
* result should be G**2mmu**5/192/pi**3
ME2=32.*(MMU**2-2*MMU*ENUE)*MMU*ENUE
RETURN
END
```

# MC: Προσομοίωση φασικού χώρου

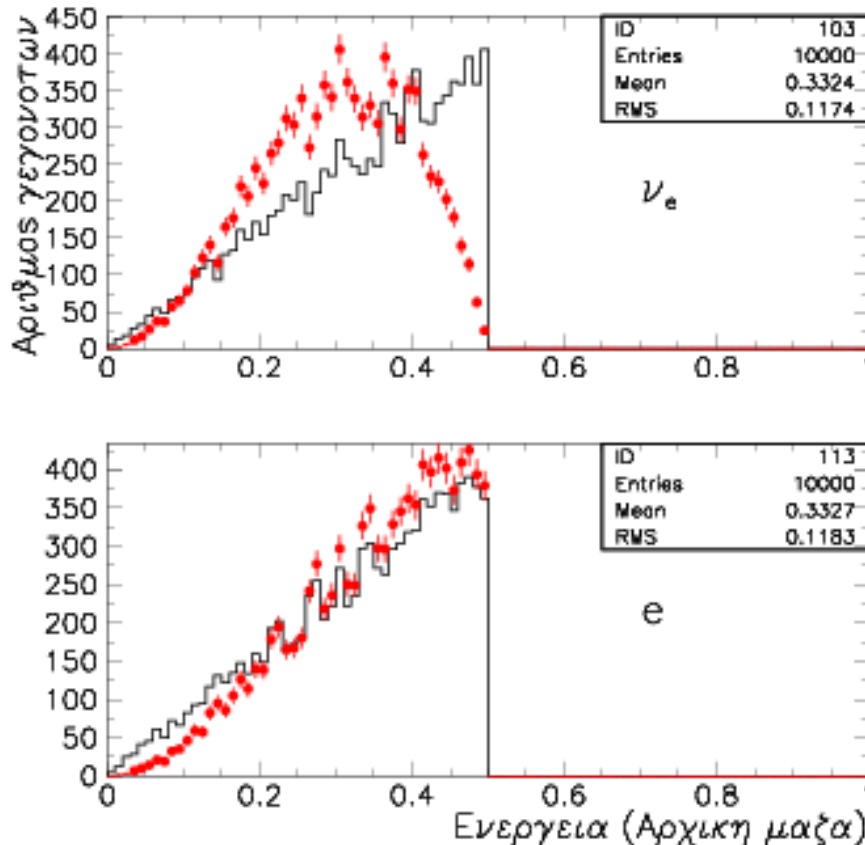
## Υπολογισμός πλάτους $\Gamma$ στη διάσπαση μιονίου



Η κατανομή προέκυψε με εφαρμογή μεθόδου ολοκλήρωσης Monte Carlo γεννήτορα τύπου M, 1000 φορές. Κάθε δείγμα αποτελούνταν από 10000 γεγονότα

(Πραγματική τιμή ολοκληρώματος – εκτιμώμενη τιμή ολοκληρώματος) / εκτιμώμενο σφάλμα.

# MC: Προσομοίωση φασικού χώρου



## Παρατηρήσεις:

- 1) Παρόλο που στην προσομοίωση το  $e$  ήταν το σωματίδιο 2 και το  $\nu_e$  το σωματίδιο 4 **Σε επίπεδο φασικού χώρου το  $e$  και το  $\nu_e$  είναι ίδια**
- 2) **Κινηματικά** ο αλγόριθμος συμπεριφέρεται σωστά (**άνω όριο τη μισή μάζα του μιονίου**)
- 3) **Στατιστικά** ο αλγόριθμος συμπεριφέρεται σωστά (**περισσότερες καταστάσεις σε μεγάλες ορμές**)
- 4) **Η διαφοροποίηση** επέρχεται με την εισαγωγή του δυναμικού όρου (**στάθμιση με το τετράγωνο του στοιχείου πίνακα**)