Applied Computational & Numerical Methods

Xinchou Lou

Chinese Academy of Sciences School of Graduate Studies

&

the Institute of High Energy Physics

Lecture IV

June 27, 2011

Applied Computational & Numerical Methods

Lecture IV

- ❑ Fits & regressions
- ❑ Examples
- ❑ The Monte Carlo Simulation
- ❑ Project 3

Project Schedule

Applied Computational & Numerical Methods

Week 1 **Getting started with the ROOT program package**

Introduction to the program, installation, setup, running, macros and document Tutorial

Weak 2 **Fits and the regression**

Basic assignment: Fit of functions to data: parameter determination and the goodness of the fit

Advanced assignment: Measurement of the lifetimes of heavy flavored hadrons

Week 3 **Monte Carlo random variates; Monte Carlo experiments**

Basic assignment: Random number generation with root, statistical features, confidence intervals

Advanced assignment: A Monte Carlo based, statistical experiment to determine the significance of an observation

Project Schedule

Applied Computational & Numerical Methods

Week 4 **Numerical methods**

Partial differential equations

Week 5 **Neural network method**

Basic assignment: Backprop training on data, test of training results, optimization of the forecast capability

Advanced assignment: An optimization for new particle search

Week 6 **Project presentations**

Fit & Regression

To use the available information/data to the maximum extent in a fi
the maximum likelihood fit provides a very powerful approach to
parameter determination, though it does not answer the question if t
data-theory/model would match.

<u>The Method</u>

$$L(\vec{p}\,|\,x_1, x_2, ..., x_n) = \prod \ln \mathrm{Pr}_i(\vec{p}\,|\,x_i)$$

where x_i are data points, and $\vec{p}$ is the set of parameters to be determined in the fit.

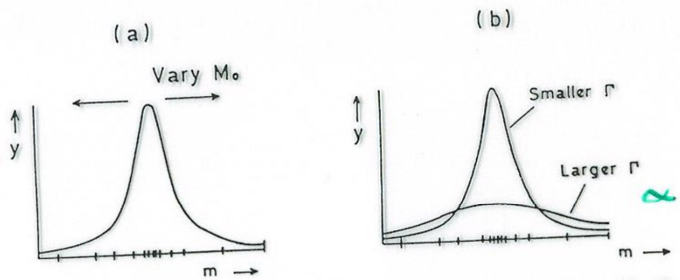
An equivalent expression, known as the log likelihood

$$\ln L = \sum \ln \mathrm{Pr}_i(\vec{p}\,|\,x_i)$$

ML maximize L or $\ln L$ to extract the parameters, and the errors on the are evaluated to be a 68% confidence contour space between the fitted $\vec{p}$ and $\vec{p} \pm \delta_p$ with $\ln L = \ln L_{max} - ½$.

Fit & Regression

Maximum Likelihood Fit $\mathcal{L}(m_0, \Gamma)$



$$\propto \frac{\Gamma}{(m-m_0)^2+\frac{\Gamma^2}{4}} \cdot \frac{1}{2\pi} = y_i(m_0,\Gamma)$$

Fig. 4.4. A visual representation to show how the maximum likelihood method succeeds in finding the best value of the parameters. The bars along the m-axis represent experimental measurements of a set of mass values m_i, which are fitted by a Breit–Wigner resonance shape. In (a) the width of the resonance is kept fixed and the central mass M_0 is varied. This has the effect of sliding the $y(M_0, \Gamma)$ curve along the m-axis, while keeping its height and shape unaltered. For a given position of the curve, we multiply together all the y_i values (i.e. y evaluated at each of the m_i) to obtain the likelihood function $\mathscr{L}$. The value of M_0 which maximises $\mathscr{L}$ is then the best estimate of the mass of the state. This clearly occurs when the curve is centred on the concentration of experimental mass values. In (b), the mass of the state is kept constant but its width is varied. Since the curves are normalised to constant area, the curve for smaller Γ is higher at the central value than is that for larger Γ. The best value of Γ is that for which the product of the y_i-values is largest. In this case it is the spread in the experimental mass values which is effective in determining Γ.

<u>Error Evaluation</u>

The log likelihood $\ell = \ln L = \sum \ln \Pr_i(\vec{p}\,|\,x_i)$ can be expanded in a Taylor series in an approximation

$$\ell = \ell_{\max} + \frac{1}{2}\ell'' \delta_p^2 + ...$$
$$= \ell_{\max} + \frac{1}{2c}\delta_p^2 + ... \quad (\mathrm{c}=\frac{1}{\ell''})$$

Considering the case where the likelihood is well behaved. Around the true value for a parameter p_i the distribution is Gaussian like

$$\ell \sim \ln\{e^{-\frac{(p_i - p_{0i})^2}{2\sigma^2}}\}$$

The second derivative of the likelihood at p_{0i} is

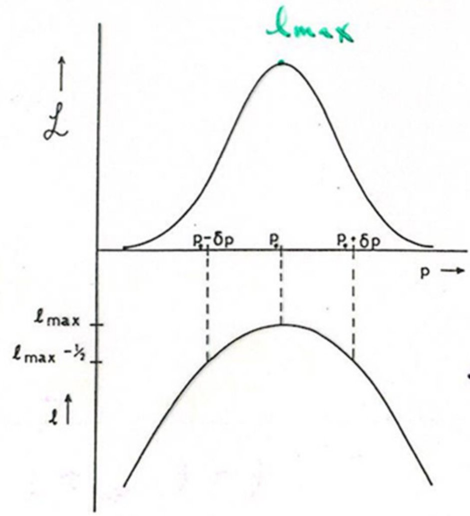
$$\ell'' \sim \frac{-1}{\sigma^2} \quad (\text{or } c \sim \sigma^2)$$

The errors on a parameter p_i correspond to the variations in its value when the log likelihood is reduced or increased by ½:

$$\ell = \ell_{\max} - \frac{1}{2} \quad \Rightarrow \; \delta_p \sim \begin{matrix} +\sigma_{p+} \\ -\sigma_{p-} \end{matrix}$$

Fit & Regression

Maximum likelihood method



$$\mathcal{L} = \prod_{i=1}^{n} y_i(m_0, \Gamma)$$

$$\ell = \log \mathcal{L} = \sum \log y_i$$

$$\ell = \ell_{max} + \frac{1}{2}\ell'' \delta p^2 + \dots$$

$$\mathcal{L} \sim \exp\left(-\frac{\delta p^2}{2\sigma^2}\right)$$

$$\ell'' = \frac{\partial \ell}{\partial p}\Big|_{\delta p=0} \sim \frac{1}{\sigma^2}$$

$$\ell = \ell_{max} \pm \frac{1}{2} \Rightarrow P_0 \pm \sigma_p$$

Fig. 4.5. In many situations, ℓ (the logarithm of the likelihood function) tends to be parabolic as a function of the parameter p in th region of the maximum. In that case, the likelihood function $\mathscr{L}$ is Gaussian distributed. The best estimate of the parameter p_0 is that which maximises either $\mathscr{L}$ or ℓ. The accuracy δp with which p_0 is determined is defined by the condition $\ell(p_0 \pm \delta p) = \ell(p_0) - \frac{1}{2}$. This is here equivalent to the statement that δp^2 is the variance of the $\mathscr{L}$ distribution.

Fit & Regression

d technique to extract errors in a maximum likelihood fit. Note that the errors on the + and – sides are not always the same.

The error matrix of the fit for the parameters $\vec{p}$ is M, its elements are also determined from the log likelihood ℓ:

$$(M_{ij})^{-1} = \left(-\frac{\partial^2 y}{\partial p_i \partial p_j} \right)$$

Fit & Regression

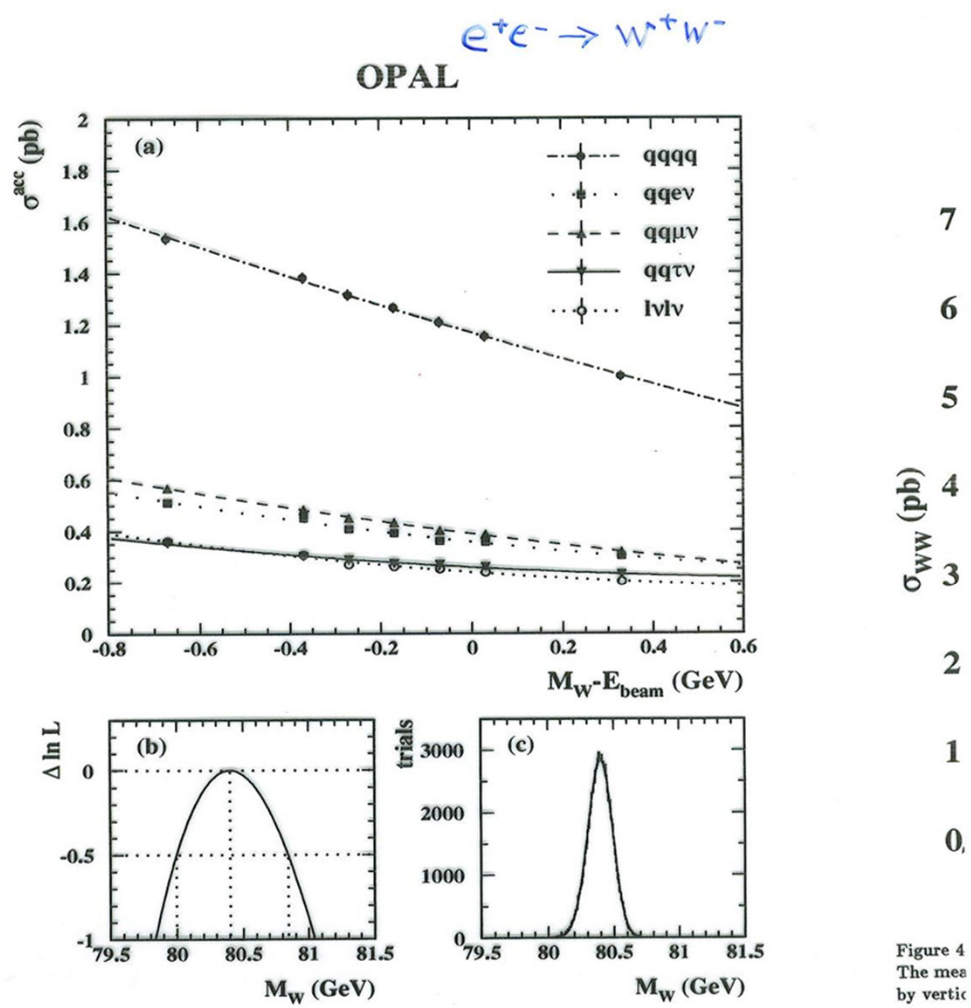


Figure 3: The accepted cross-sections in each W^+W^- decay channel as a function of $M_W - E_{beam}$ are illustrated in (a), using the event generators described in the text. In each case, this is parametrised by a second order polynomial. The likelihood function and the corresponding statistical uncertainty are shown in (b) for $\sqrt{s} = 161.3$ GeV. Plot (c) shows the distribution of M_W values evaluated using repeated Monte Carlo trials. Its width gives the systematic uncertainty.

17

Fit & Regression

he standard technique to extract errors in a Chi-square fit.

The error matrix of the fit for the parameters $\vec{p}$ is M, its elements are also determined from the χ^2 variable:

$$(M_{ij})^{-1} = \left(\frac{1}{2} \frac{\partial^2(\chi^2)}{\partial p_i \partial p_j} \right) \quad \text{or} \quad (M_{ij}) = \left(\frac{1}{2} \frac{\partial^2(\chi^2)}{\partial p_i \partial p_j} \right)^{-1}$$

A closer look at matrix elements of the IEM:

$$\begin{aligned} \chi^2(\vec{p}) &= \sum_{i=1}^{n} \frac{(X_i^{\exp} - X_i^{theory}(x,\vec{p}))^2}{\sigma_i^2} \quad \text{(a random variables depending on } p_i\text{)} \\ &\cong -2\ln(L) + \text{constant} \\ &= \frac{p_1^2}{\sigma_{p_1}^2} + \frac{p_2^2}{\sigma_{p_2}^2} + \ldots + \frac{p_n^2}{\sigma_{pn}^2} + \text{constant} \end{aligned}$$

Using the Gaussian approximation, at or near the true value for p_i, i contribution to $\chi 2$ is

$$\frac{(p_i - p_{i0})^2}{\sigma_{p_i}^2}$$

Therefore $\frac{\partial}{\partial p_i}(\chi^2) = \frac{2(p_i - p_{i0})}{\sigma_{p_i}^2}$, and $\frac{\partial^2}{\partial p_i^2}(\chi^2) = \frac{2}{\sigma_{p_i}^2}$

Fit & Regression

Examples

The likelihood:

$$l = -\frac{1}{4}(13x^2 + 6\sqrt{3}xy + 7y^2)$$

To maximize the likelihood

$$\frac{\partial l}{\partial x} = \frac{1}{4}(26x + 6\sqrt{3}y) = 0$$
$$\frac{\partial l}{\partial y} = \frac{1}{4}(6\sqrt{3}x + 14y) = 0$$

Solve this linear equations, we obtain:

$$x = 0$$
$$y = 0$$

And the error matrix is

$$M_{ij} = -\left(\frac{\partial^2 l}{\partial p_i \partial p_j}\right)^{-1} = -\begin{pmatrix} \frac{\partial^2 l}{\partial x^2} & \frac{\partial^2 l}{\partial x \partial y} \\ \frac{\partial^2 l}{\partial y \partial x} & \frac{\partial^2 l}{\partial y^2} \end{pmatrix}^{-1} = \frac{2}{64}\begin{pmatrix} 7 & -3\sqrt{3} \\ -3\sqrt{3} & 13 \end{pmatrix}$$

Fits & Regression

A closer look at matrix elements of the IEM:

$$\chi^2(\vec{p}) = \sum_{i=1}^{n} \frac{(y_i^{\exp} - y_i^{theory}(x,\vec{p}))^2}{\sigma_i^2} \quad \text{(a random variables depending on } \mathrm{p_i}\text{)}$$

$$\cong -2\ln(L) + \text{constant}$$

$$= \frac{\mathrm{p}_1^2}{\sigma_{p_1}^2} + \frac{\mathrm{p}_2^2}{\sigma_{p_2}^2} + ... + \frac{\mathrm{p}_\mathrm{n}^2}{\sigma_{pn}^2} + \text{constant}$$

Using the Gaussian approximation, at or near the true value for p_i, i contribution to $\chi 2$ is

$$-\frac{(p_i - p_{i0})^2}{\sigma_{p_i}^2}$$

Therefore $\frac{\partial}{\partial p_i}(\chi^2) = -\frac{2(p_i - p_{i0})}{\sigma_{p_i}^2}$, and $\frac{\partial^2}{\partial p_i^2}(\chi^2) = -\frac{2}{\sigma_{p_i}^2}$

Fits & Regression

□ **More on errors associated with the ChiSquare Fit**

To extract the errors and co-variances we define an inverse error ma

$$H = \begin{pmatrix} \frac{\partial^2}{\partial p_1^2}\left(\chi^2\right) & \cdots\cdots & \frac{\partial^2}{\partial p_1 \partial p_n}\left(\chi^2\right) \\ . & . & . \\ \frac{\partial^2}{\partial p_n \partial p_1}\left(\chi^2\right) & \cdots\cdot & \frac{\partial^2}{\partial p_n^2}\left(\chi^2\right) \end{pmatrix}$$

The error matrix for the set of parameters p_i is obtained by invertinǵ
H matrix

$$M = H^{-1} \quad (\mathrm{H}_{ij} = \left(\frac{1}{2}\frac{\partial^2(\chi^2)}{\partial p_i \partial p_j}\right))$$

Fits & Regression

□ Error Contour

With multiple parameters fit, the parameters are normally correlatec Instead of clear linear intervals for the ranges for the parameters, a closed contour defines the 1 standard error boundary as a function c parameters involved. This is achieved by requiring

$$\chi^2_{\min} + 1$$

Example

Assume

$$\chi^2_{xy} = \chi^2_{\min} + \frac{\delta x^2}{\sigma_x^2} + \frac{\delta y^2}{\sigma_y^2}$$

The error contour is then

$$\frac{\delta x^2}{\sigma_x^2} + \frac{\delta y^2}{\sigma_y^2} = 1$$

Keep one fixed at the fitted value and allow the other to float. The amount of change in the other variable is exactly the standard error the variable.

Fits & Regression

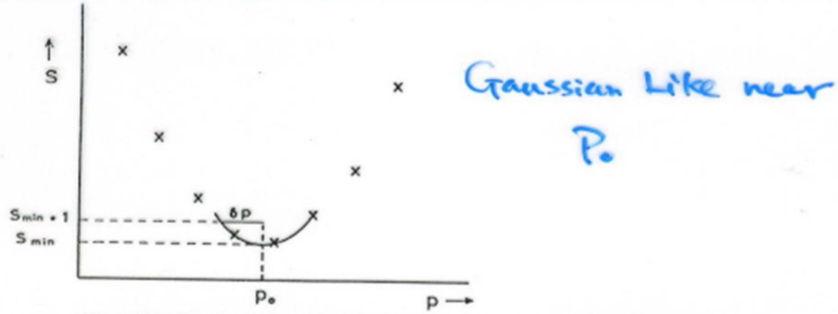


Fig. 4.14. Do-it-yourself minimisation, for a problem involving a single parameter p. The sum of squares S is calculated at a series of values of the parameter p; these are represented by the crosses. The minimum is then obtained either by inspection, or by a simple computer subroutine which calculates the parabola (solid curve) through the three points nearest the minimum. The minimum of the parabola gives the best estimate p_0 of the parameter, and the width of the parabola determines the error δp (assumed to be symmetric).

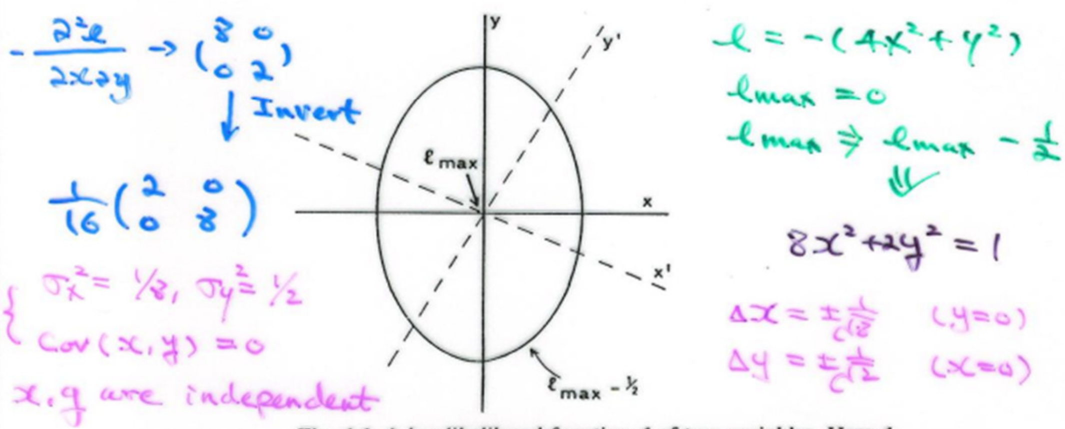


Fig. 4.6. A log-likelihood function ℓ of two variables. Here ℓ maximises at the origin. The ellipse shown corresponds to the contou for $\ell_{max} - \frac{1}{2}$, which is used to obtain the errors on the variables x anc y. Since the axes of the ellipse are parallel to the x and y axes, the errors are uncorrelated. The simplest way to produce the effect of correlations is to rotate the axes. Then ℓ still maximises at the origin, but we see that the ellipse is now inclined. As y' increases from its optimum value (zero), the value of x' required to maximise ℓ decreases; this is a negative correlation. The magnitudes of the error: (including their possible correlation) are expressed in terms of the error matrix, which is derived from the likelihood function by equations (4.33′) and (4.33).

Fits & Regression

Kinematic fit with constraints:

In many cases measured quantities can be improved by making use known physics laws and other conditions.

Consider an elastic collision in space between equal mass objects

$$p + \mathrm{p} \rightarrow \mathrm{p} + p$$

<u>ematic constraints:</u> $C = \theta_1 + \theta_2 - \pi/2 = 0$

Fits & Regression

ıne the chisquare variable

$$S=\frac{(\theta_1^m-\theta_1)^2}{\varepsilon^2}+\frac{(\theta_2^m-\theta_2)^2}{\varepsilon^2}=\frac{(\theta_1^m-\theta_1)^2}{\varepsilon^2}+\frac{(\theta_2^m-\frac{\pi}{2}+\theta_1)^2}{\varepsilon^2}$$

re ε is the error on the measured angles $\theta^m{}_1$ and $\theta^m{}_2$. By requiring

$$\frac{\partial S}{\partial\theta_1}=\frac{-2(\theta_1^m-\theta_1)}{\varepsilon^2}+\frac{2(\theta_2^m-\frac{\pi}{2}+\theta_1)}{\varepsilon^2}=0$$

ıolution for θ_1 and θ_2 are found to be

$$\theta_1=\theta_1^m+\frac{1}{2}(\frac{\pi}{2}-\theta_1^m-\theta_2^m),\ \theta_2=\theta_2^m+\frac{1}{2}(\frac{\pi}{2}-\theta_1^m-\theta_2^m)$$

errors on the improved angle measurements are

$$\sigma_\theta^2=\left(\frac{\partial\theta_1}{\partial\theta_1^m}\right)^2\varepsilon^2+\left(\frac{\partial\theta_1}{\partial\theta_2^m}\right)^2\varepsilon^2=\left(\frac{1}{2}\right)^2\varepsilon^2+\left(\frac{1}{2}\right)^2\varepsilon^2=\frac{\varepsilon^2}{2}$$

$$\Rightarrow\ \sigma_\theta=\frac{\varepsilon}{\sqrt{2}}$$

Fits & Regression

minimized chisquare value is

$$S_{\min} = \frac{(\frac{\pi}{2} - \theta_1^m - \theta_2^m)^2}{2\varepsilon^2}$$

es:

I. Error reduced by 28% after the kinematic fit.

I. Simple, straight fit – linear in θ_1 and θ_2.

II. Though $\theta^m{}_1$ and $\theta^m{}_2$ are independent, θ_1 and θ_2 are now full correlated.

r matrix

$$M_m = \begin{pmatrix} \varepsilon^2 & 0 \\ 0 & \varepsilon^2 \end{pmatrix} \text{ for } \theta_1^m \text{ and } \theta_2^m$$

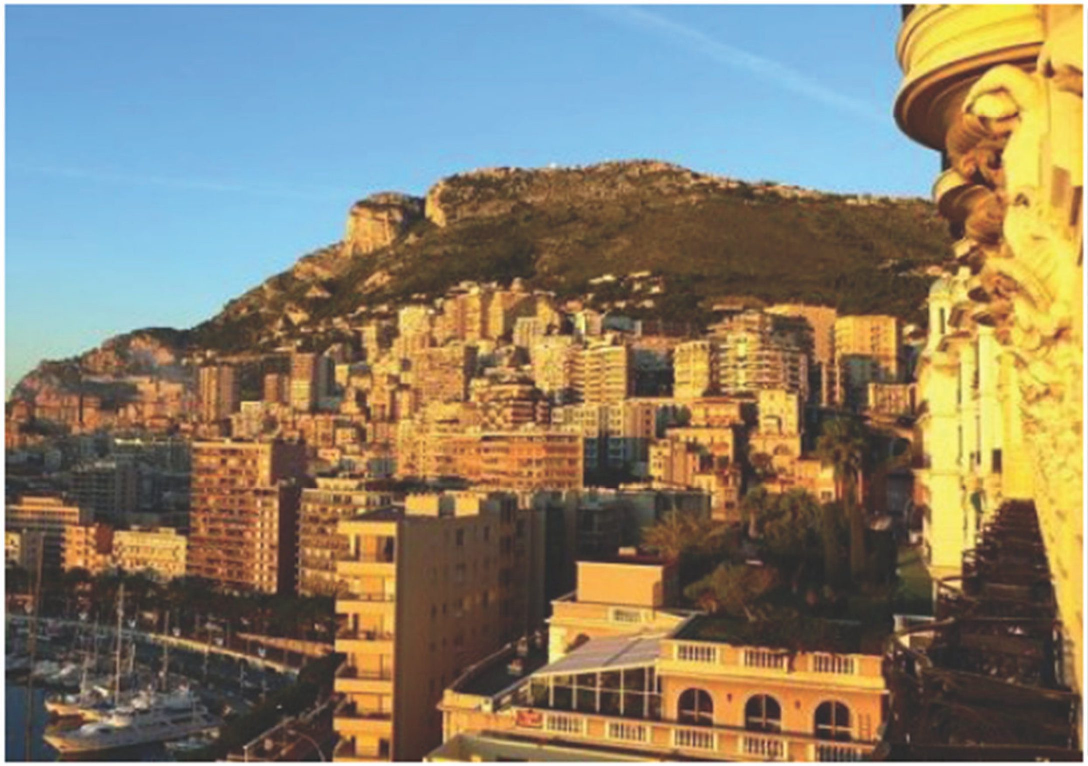
new error matrix for θ_1 and θ_2 is

$$M_{fit} = \tilde{T} M_m T = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \varepsilon^2 & 0 \\ 0 & \varepsilon^2 \end{pmatrix} \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

$$= \frac{\varepsilon^2}{2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

Monte Carlo Simulation

What is Monte Carlo?



Nick Whiteley 2010

Monte Carlo Simulation

What is the use of the Monte Carlo Simulation ?

I heard or read over the years:

" **is an brut force approach to problems which have random variables and the solution can not be expressed analytically**. "

" **is widely used for parameter analyses**."

For example: to know the influence of a parameter, say vapor permeability of a wall layer, one obtain a number of results, say the total moisture accumulation. Each result is obtained by a randomly selected permeability. The selection is according to a propabality density distribution (such as normal or uniform). The distribution of the resuls is the influence by the parameter variation.

Monte Carlo Simulation

" is the last resort to understand a complex system quantitatively."

For example: multi-body quantum system in which large number of states are not fully populated; rather their occupancies are described by the probability density functions (PDF). A real time description of such systems can only be achieved using the Monte Carlo simulation method.

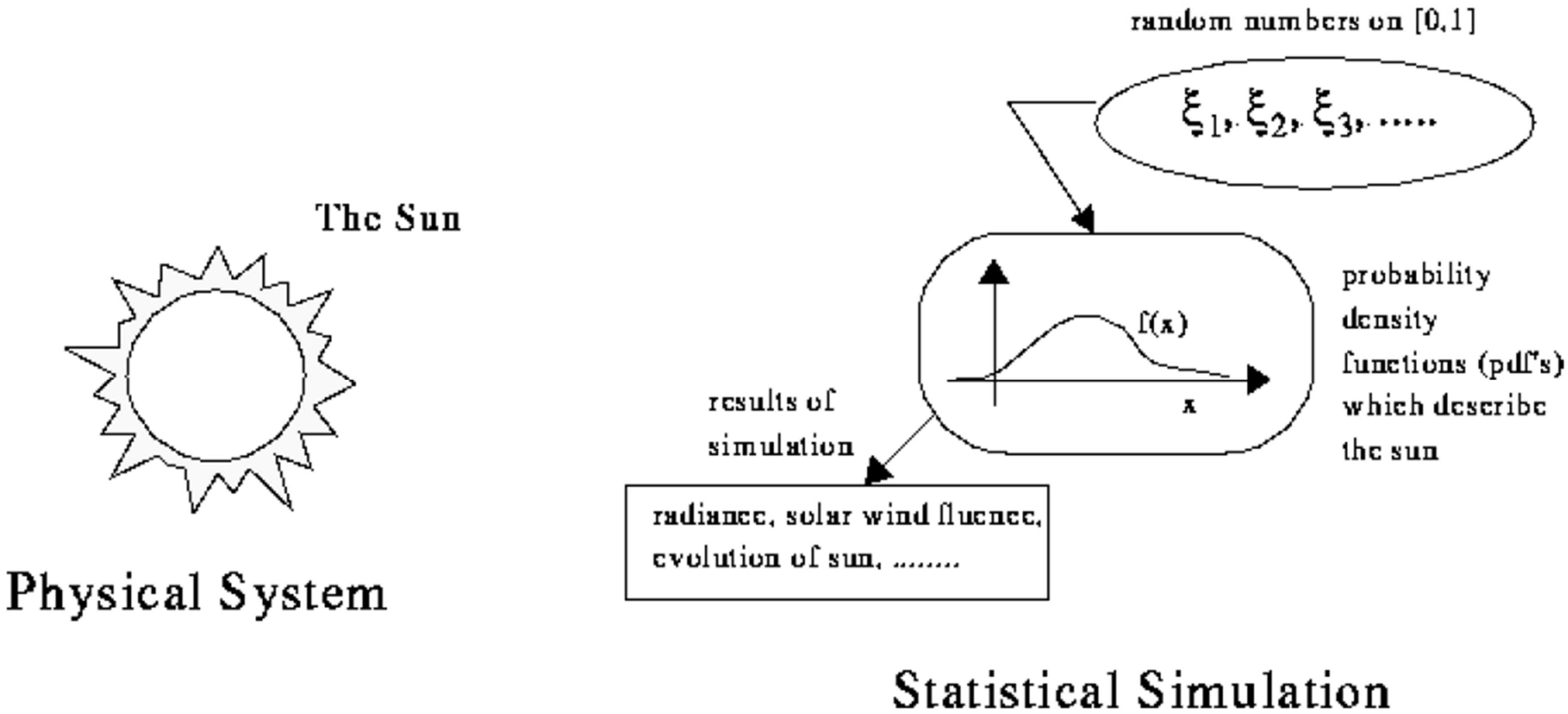
" can be described as a statistical simulation method that utilizes sequences of random numbers to simulate the processes often too complicated to described analytically."

Monte Carlo method as a scientific technique gained its status of capable of addressing the most complex applications. The term ``Monte Carlo'' was coined during the Manhattan Project of World War II, because it involves randomness or the game of chance, and because the city of Monte Carlo, the capital of Monaco, was a center for gambling and similar pursuits. Monte Carlo is now used routinely in many diverse fields.

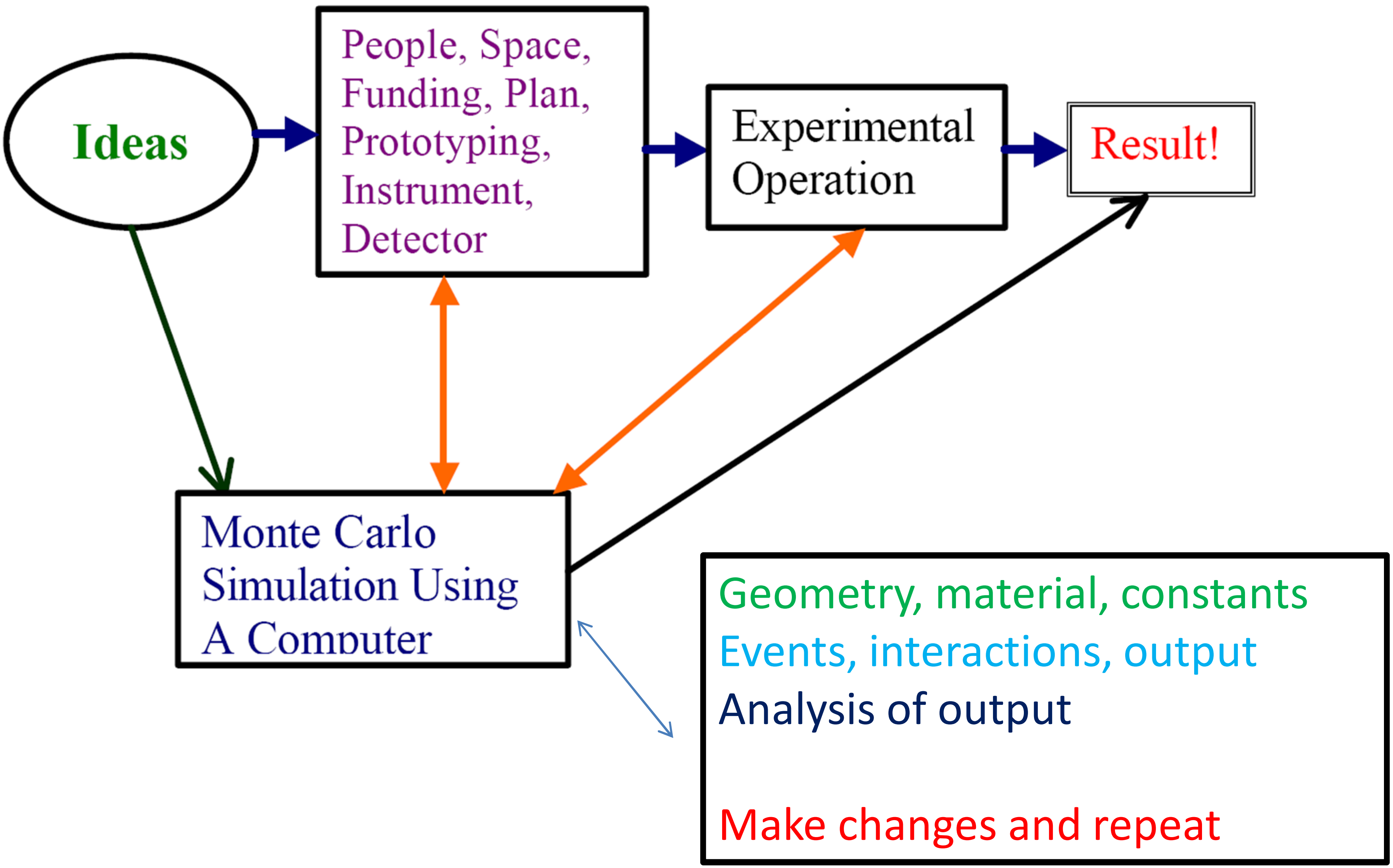
Monte Carlo Simulation

" is, in many occasions, the cheapest or only way to do the experiment."

Many experiments or systems are too expensive to build or to optimize. Monte Carlo simulation method makes it possible to carry out these experiments or setting up the systems on a computer and can run them many times without occurring high cost.



Monte Carlo Simulation



Monte Carlo Simulation

Nuclear Interactions

Quantum Physical Systems

Detection System

Cosmology and Astrophysics, **Space Research**

Large Degrees-of-Freedom Physical Systems

Traffic Flow and Control

Biological Systems

Economics, **Financial Markets**, Effects of Economic Policies

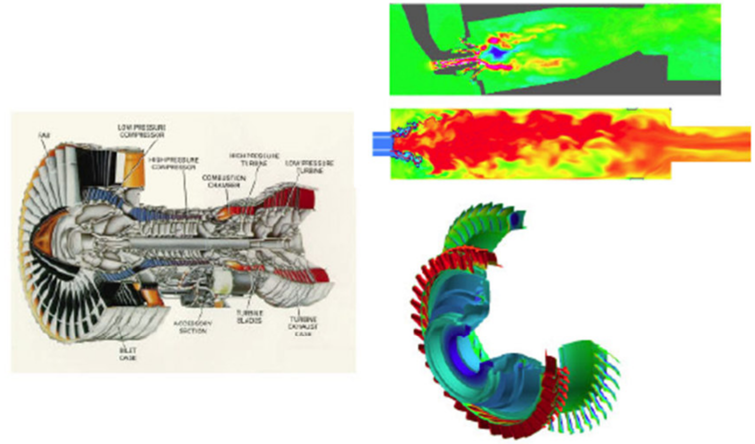
Seismology and Oil Exploration

System Designs

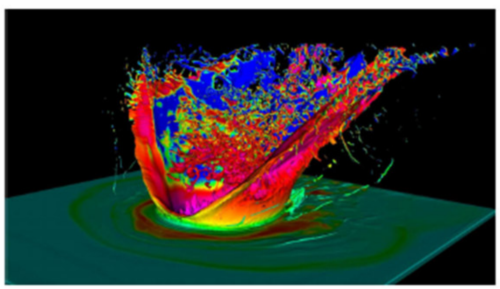
……. Many More

Monte Carlo Simulation

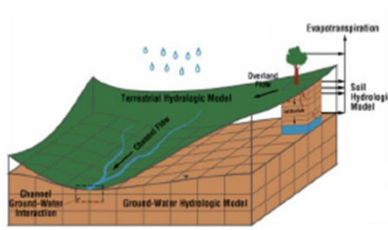
- The growth of computing power over the last 50 years has enabled us to address and “solve” many important technical problems for society
- Codes contain Realistic models, good spatial and temporal resolution, realistic geometries, realistic physical data, etc.



Stanford ASCI Alliance—Jet Engine Simulation



G. Gisler et al-Impact of Dinosaur Killer Asteroid

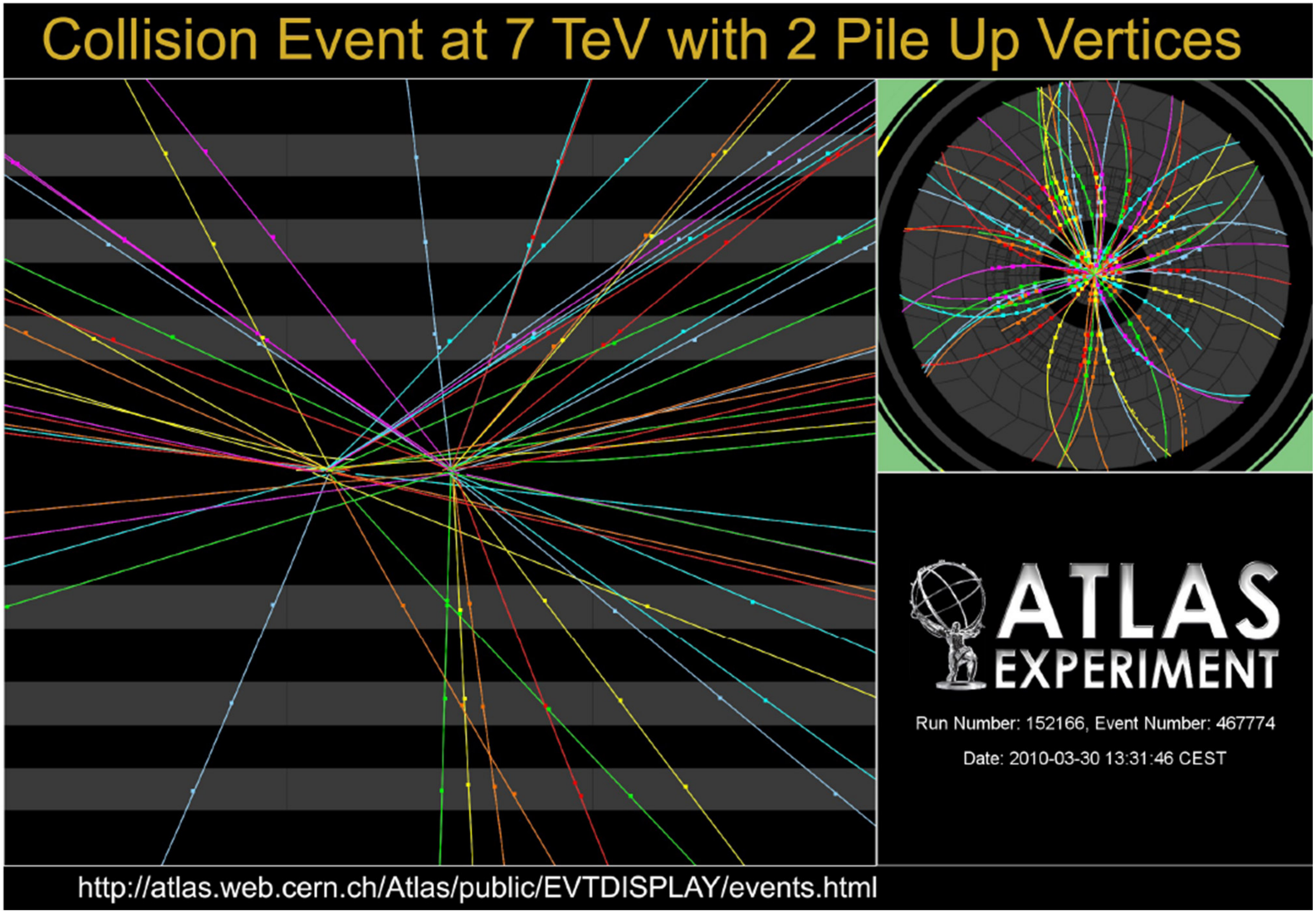


L. Winter et al-Rio Grande Watershed

U. Of Illinois ASCI Alliance—Shuttle Rocket Booster Simulation

2

Monte Carlo Simulation



Monte Carlo Simulation

Simplistic View of a Monte Carlo Simulation System

(1) Random number generation

(2) Probability Density Functions (PDF)

(3) Mapping random numbers into the (PDF) space

(4) Sampling of generated events

(5) Comparison to prototype system or data or theoretical expectations

(6) Calibration, correction or best estimate

(7) Statistical analysis of the simulation: intermediate or final results

(8) Error analysis

(9) Optimization with error reduction, parameter variation, etc.

Monte Carlo Simulation

One of many definitions

A Monte Carlo method consists of

- "representing the solution of a problem as a parameter of a hypothetical population, and
- using a random sequence of numbers to construct a sample of the population, from which statistical estimates of the parameter can be obtained."

(Halton, 1970)

Sometimes referred to as *stochastic simulation*.

Nick Whiteley 2010

Introduction – simple examples

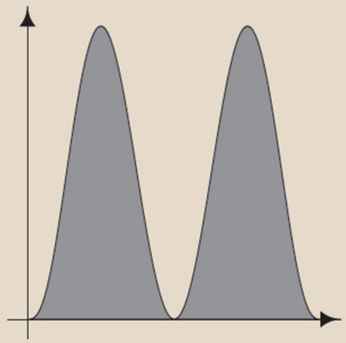
Examples of applications of Monte Carlo methods (1)

Numerical Integration

Objective is to estimate an integral

$$\int_{\mathcal{X}} f(\mathbf{x})\, d\mathbf{x},$$

which is analytically intractable.



N. Whiteley

Introduction – simple examples

Examples of applications of Monte Carlo methods (2a)

Bayesian statistics

- Data $\mathbf{y}_1, \ldots, \mathbf{y}_n$ and model $f(\mathbf{y}_i|\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is some parameter of interest.

  $\rightsquigarrow$ Likelihood $l(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\theta}) = \prod_{i=1}^{n} f(\mathbf{y}_i|\boldsymbol{\theta})$
- Frequentist estimate of $\boldsymbol{\theta}$ is the maximiser of $l(\mathbf{y}_1, \ldots, \mathbf{y}_n)$ ("maximum likelihood estimate").
- In the frequentist framework $\boldsymbol{\theta}$ is a parameter, not a random variable.

N. Whiteley

Introduction – simple examples

Examples of applications of Monte Carlo methods (2b)

Bayesian statistics (continued)

- In the Bayesian framework $\boldsymbol{\theta}$ is a random variable with prior distribution $f^{\text{prior}}(\boldsymbol{\theta})$. After observing $\mathbf{y}_1, \ldots, \mathbf{y}_n$ the posterior density of f is

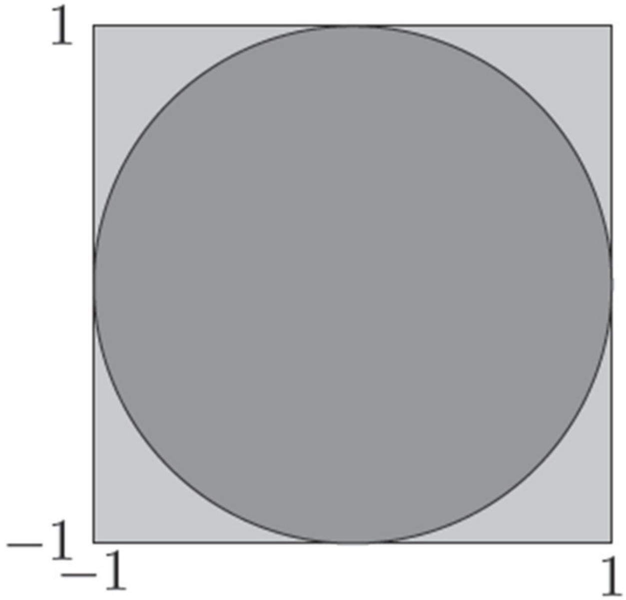
$$
\begin{aligned}
f^{\text{post}}(\boldsymbol{\theta}) &= f(\boldsymbol{\theta}|\mathbf{y}_1, \ldots, \mathbf{y}_n) \\
&= \frac{f^{\text{prior}}(\boldsymbol{\theta}) l(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\theta})}{\int_\Theta f^{\text{prior}}(\boldsymbol{\vartheta}) l(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\vartheta})\, d\boldsymbol{\vartheta}} \\
&\propto f^{\text{prior}}(\boldsymbol{\theta}) l(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\theta})
\end{aligned}
$$

- For many complex models the integral in the denominator is hard to compute
$\rightsquigarrow$ use of a Monte Carlo approximation

N. Whiteley

Example 1.1: Raindrop experiment for computing π (1)

- Consider "uniform rain" on the square $[-1,1]\times[-1,1]$, i.e. the two coordinates $X, Y \overset{\text{i.i.d.}}{\sim} \mathsf{U}[-1,1]$.
- Probability that a rain drop falls into the dark circle is



$$\mathbb{P}(\text{drop within circle}) = \frac{\text{area of the unit circle}}{\text{area of the square}} = \frac{\displaystyle\iint_{\{x^2+y^2\le 1\}} 1\,dxdy}{\displaystyle\iint_{\{-1\le x,y\le 1\}} 1\,dxdy} = \frac{\pi}{2\cdot 2} = \frac{\pi}{4}.$$

Nick Whiteley 2010

Example 1.1: Raindrop experiment for computing π (2)

- If we know π, we can compute $\mathbb{P}(\text{drop within circle}) = \frac{\pi}{4}$.
- Consider n independent raindrops, then the number of rain drops Z_n falling in the dark circle is a binomial random variable:

$$Z_n \sim \mathsf{B}(n, \theta), \qquad \text{with } \theta := \mathbb{P}(\text{drop within circle}).$$

- We can estimate θ by

$$\hat{\theta}_n = \frac{Z_n}{n}.$$

- Thus we can estimate π by

Nick Whiteley 2010

$$\hat{\pi}_n = 4\hat{\theta}_n = 4 \cdot \frac{Z_n}{n}.$$

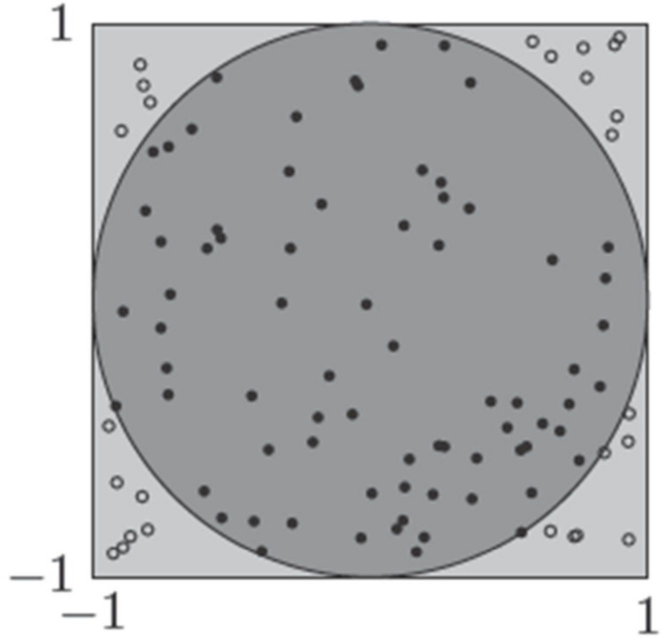
Monte Carlo Simulation

- Result obtained for $n = 100$ raindrops: 77 points inside the dark circle.
- Resulting estimate of π is

$$\hat{\pi} = \frac{4 \cdot Z_n}{n} = \frac{4 \cdot 77}{100} = 3.08,$$

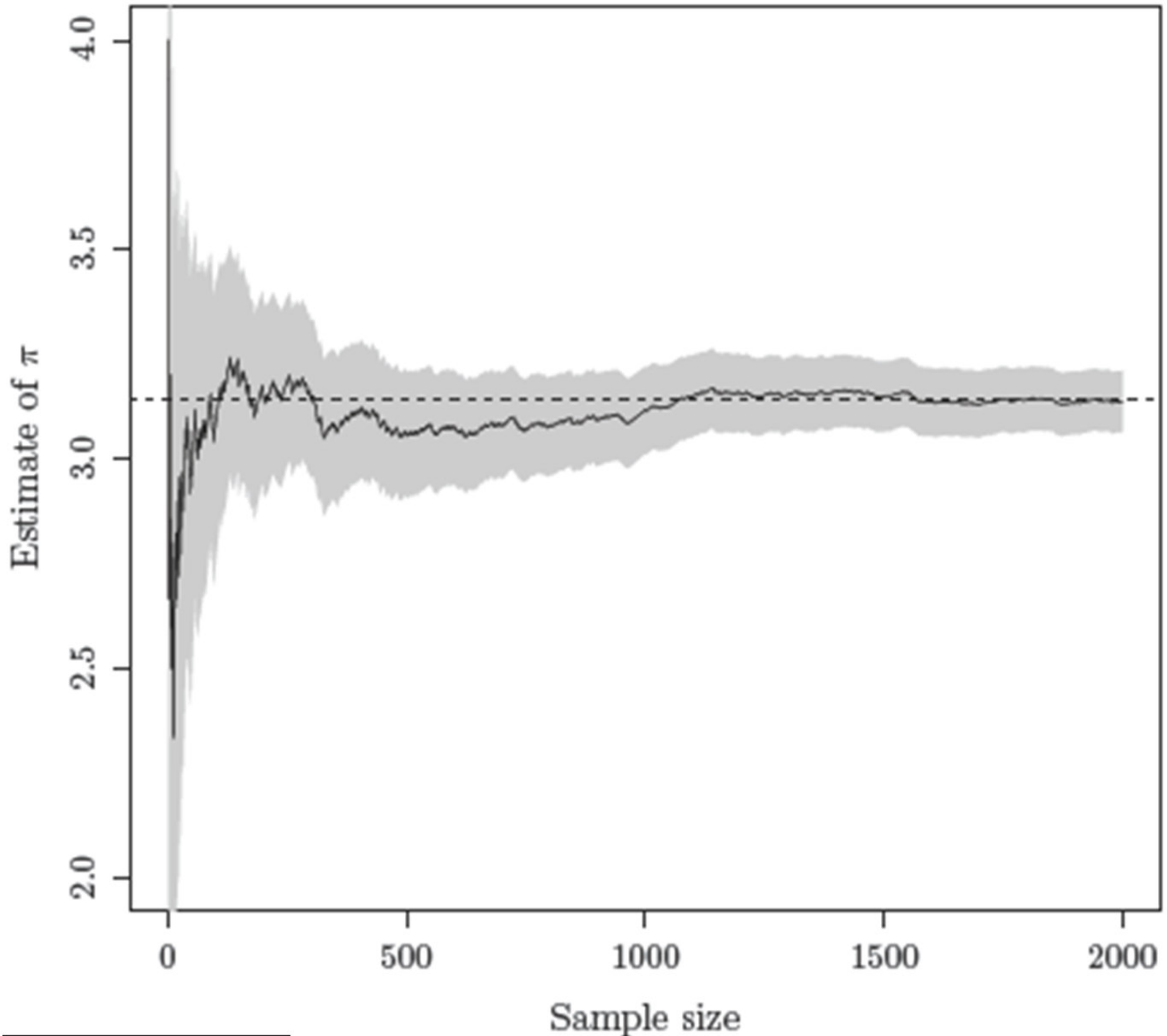
(rather poor estimate)

- However: the *law or large numbers* guarantees that $\hat{\pi}_n = \frac{4 \cdot Z_n}{n} \to \pi$ almost surely for $n \to \infty$.



Nick Whiteley 2010

Monte Carlo Simulation



Nick Whiteley 2010

Monte Carlo Simulation

Algorithm 1.1: Congruential pseudo-random number generator

1. Choose $a, M \in \mathbb{N}$,$c \in \mathbb{N}_0$, and the initial value ("seed") $Z_0 \in \{1, \ldots M-1\}$.
2. For $i = 1, 2, \ldots$

   Set $Z_i = (aZ_{i-1} + c) \mod M$, and $X_i = Z_i / M$.

$Z_i \in \{0, 1, \ldots, M-1\}$, thus $X_i \in [0, 1)$.

Nick Whiteley 2010

Monte Carlo Simulation

Pseudorandom number generator

$x_{i+1} = (ax_i + c)$ (mod m), i=1,...,n
a multiplier
c increment
m modulus

which means that $x_{i+1} = (ax_i + c) - mk_i$

where $k_i=[(ax_i + c)/m]$

denotes the largest positive integer in (axi + c)/m

• (0-1) random number can be obtained from ui=xi/m

•Clearly that the recursive formula yields a deterministic sequence, the numbers will be periodic

especially xi < m -- the sequence contains at most m distinctive numbers (*get into a loop*)

Monte Carlo Simulation

Cosider the choice of $a = 81$, $c = 35$, $M = 256$, and seed $Z_0 = 4$.

$$
\begin{aligned}
Z_1 &= (81 \cdot 4 + 35) \mod 256 = 359 \mod 256 = 103 \\
Z_2 &= (81 \cdot 103 + 35) \mod 256 = 8378 \mod 256 = 186 \\
Z_3 &= (81 \cdot 186 + 35) \mod 256 = 15101 \mod 256 = 253 \\
&\cdots
\end{aligned}
$$

The corresponding X_i are $X_1 = 103/256 = 0.4023438$, $X_2 = 186/256 = 0.72656250$, $X_1 = 253/256 = 0.98828120$.

Nick Whiteley 2010

Basic Concepts – pseudo random numbers

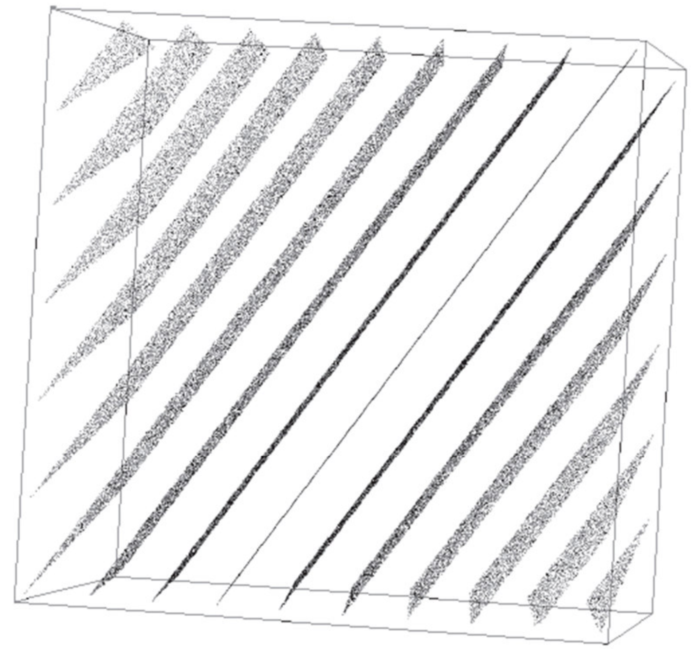
- Philosophical paradox:
  - We need to reproduce randomness by a computer algorithm.
  - A computer algorithm is deterministic in nature.

  $\rightsquigarrow$ "pseudo-random numbers"
- Pseudo-random number from $\mathrm{U}[0, 1]$ will be our only "source of randomness".
- Other distributions can be derived from $\mathrm{U}[0, 1]$ pseudo-random numbers using deterministic algorithms.

N. Whiteley

Basic Concepts – pseudo random numbers

- Very popular in the 1970s (e.g. System/360, PDP-11).
- Linear congruential generator with $a = 2^{16} + 3$, $c = 0$, and $M = 2^{31}$.
- The numbers generated by RANDU lie on only 15 hyperplanes in the 3-dimensional unit cube!



According to a salesperson at the time: "We guarantee that each number is random individually, but we don't guarantee that more than one of them is random."

N. Whiteley

Basic Concepts – pseudo random numbers

The flaw on the linear congruential generator

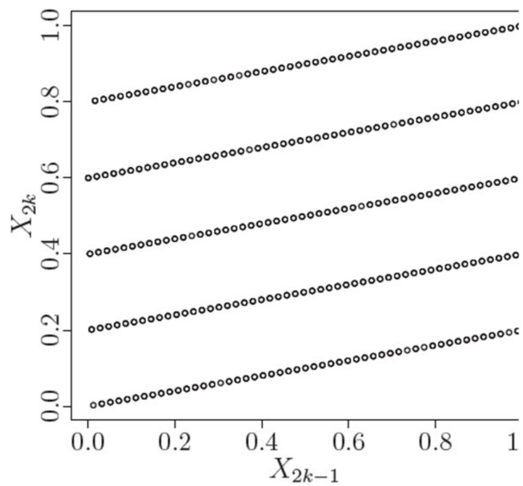
- "Crystalline" nature is a problem for every linear congurentrial generator.
- Sequence of generated values $X_1, X_2, \ldots$ viewed as points in an n-dimension cube lies on a finite, and often very small number of parallel hyperplanes.
- Marsaglia (1968): "the points [generated by a congruential generator] are about as randomly spaced in the unit n-cube as the atoms in a perfect crystal at absolute zero."
- The number of hyperplanes depends on the choice of a, c, and M.
- For these reasons do not use the linear congurential generator! Use more powerful generators (like e.g. the *Mersenne twister*, available in GNU R).

N. Whiteley

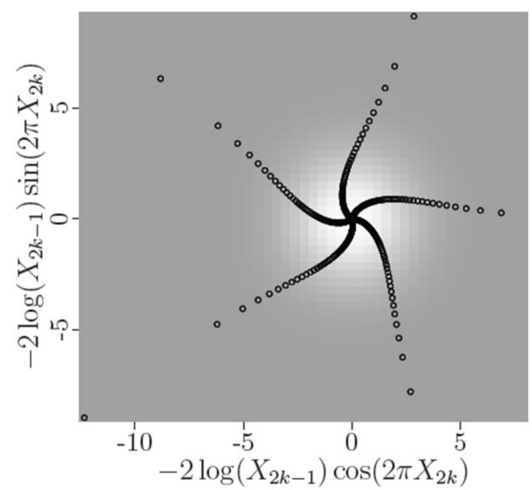
Basic Concepts – pseudo random numbers

Another cautionary example

Linear congruential generator with $a = 1229$, $c = 1$, and $M = 2^{11}$.



Pairs of generated values (X_{2k-1}, X_{2k})



Transformed by Box-Muller method

N. Whiteley

Basic Concepts – pseudo random numbers

Example Generators

<u>Literature</u> $x_0>0$, $a=2^7+1$, $c=1$, $m=2^{35}$

<u>IBM System/360</u> $x_0>0$, $a = 7^5$, $c=0$, $m=2^{31} - 1$

CERN library: $m=2^{31}-1$, $a=2^r+1$, $c=0$

ROOT Library

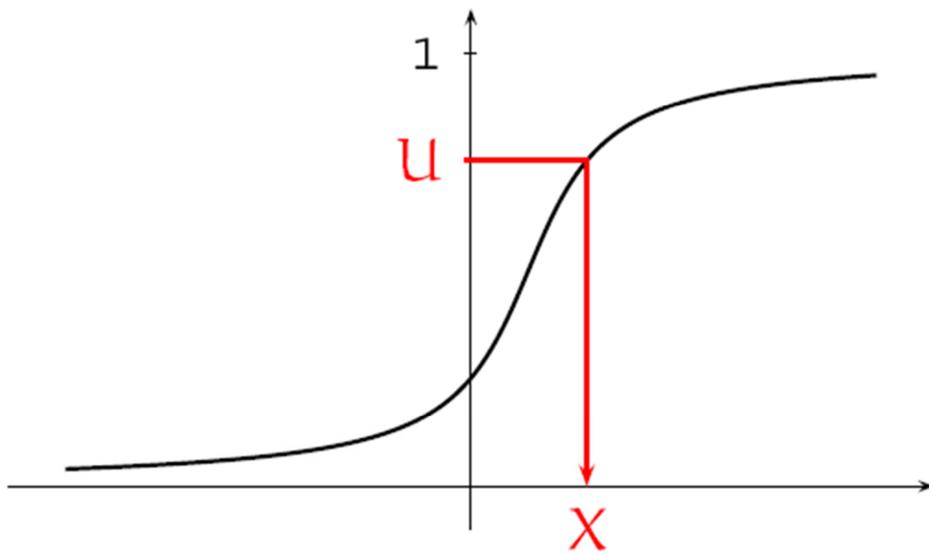
```
Float_t TRandom::Rndm(Int_t)
{
//  Machine independent random number generator.
//  Produces uniformly-distributed floating points between 0
//  Identical sequence on all machines of >= 32 bits.
//  Periodicity = 10**8
//  Universal version (Fred james 1985).

   const Float_t kCONS = 4.6566128730774E-10;
   const Int_t kMASK31 = 2147483647;

   fSeed *= 69069;
      // keep only lower 31 bits
   fSeed &= kMASK31;
      // Set lower 8 bits to zero to assure exact float
   Int_t jy = (fSeed/256)*256;
   Float_t random = kCONS*jy;
   return random;
}
```

Monte Carlo Simulation

Generation of random variates with PDF=f(x), and CDF= $F(x < x_{cutoff}) = \int_{-\infty}^{x_{cutoff}} f(x)dx$

Inversion Method // Properties

Required: (Inverse of) **CDF** F of Distribution.



$$u \sim U(0,1) \longrightarrow X = F^{-1}(u)$$

Exponential distribution: $F^{-1}(u) = -\log(1-u)$

Leydold, Hörmann & Moneta ·

Inversion Method // Properties

- The most **general** method for generating non-uniform random variates.
  Works for all distributions provided that the inverse CDF is given.
- Get **one** random variate X for each uniform U.
- **Preserves** the structural properties of the underlying uniform PRNG.

However:

- CDF and its inverse often not given in closed form.
- Need slow and/or **approximate** numerical methods.

Leydold, Hörmann & Moneta ·

Introduction – simple examples

Consider the simple integral:

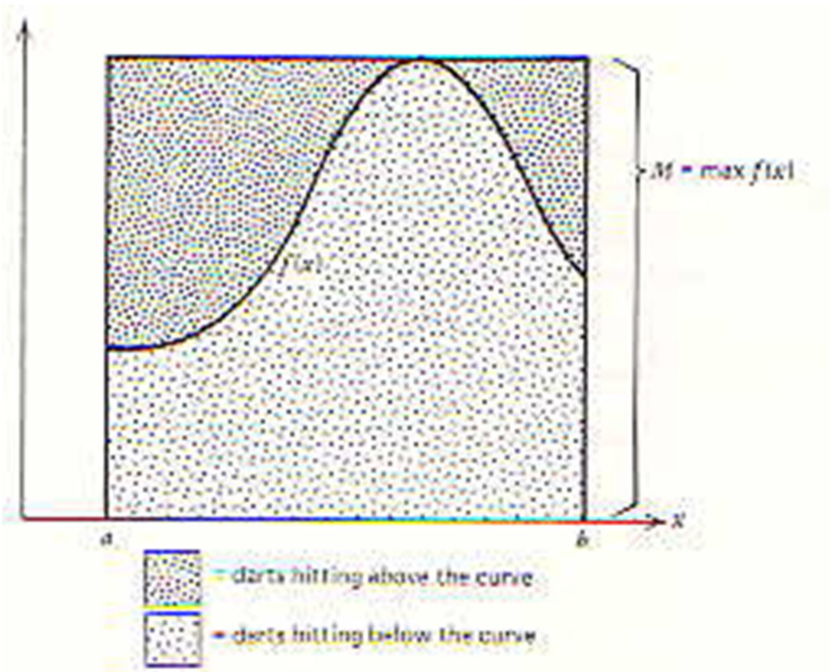
$$I = \int_b^a f(x)\, dx$$

This can be evaluated in the same way as the pi example. By randomly tossing darts at a graph of the function and tallying the ratio of hits inside and outside the function.

G. Chen

Introduction – simple examples

A Simple Integral (continued…)



- R = {(x,y): a ≤ x ≤ b, 0 ≤ y ≤ max *f(x)*}
- **Randomly tossing 100 or so darts we could approximate the integral…**
- *I = [fraction under f(x)] * (area of R)*

- **This assumes that the dart player is throwing the darts randomly, but not so random as to miss the square altogether.**

G. Chen

Project III – Introduction and discussion

Monte Carlo Random Variates & A Monte Carlo Experiment

Basic Assignment

Random number generation with root, statistical features, and confidence intervals

Advanced Assignment

A Monte Carlo based, statistical experiment to determine the significance of an observation

Project III – Introduction and discussion

Basic Assignment

Random number generation with root, statistical features,and confidence intervals

Introduction

In many scientific studies a theory can define a probability density function (PDF) of well known phenomena, PDF_{wkp}. Experimentalists can search for events that are statistically well beyond the rate predicted by PDF_{wkp}. Observation of such result may lead to important discovery and better understanding of the nature or the technical aspect of a physical system.

In physics research, commonly used PDFs include the Possion distribution

$$p(n) = \frac{<n>^n}{n!} e^{-<n>},$$

where <n> is the mean, n the total number of successes or signal events observed during the period of the experiment, and the Gaussian PDF distribution

$$f(x, \bar{x}, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}}$$

Project III – Introduction and discussion

where $\bar{x}$ is the mean of the variable x, and σ is the standard deviation. For a study involving a total of N trials, the event density function (EDF) is given by

$$n(x) = N \times f(x)$$

for a Gaussian distribution. The total number of events expected in an interval (x_l, x_h) is determined from

$$\mathrm{N} \times \sum_{k=x_l}^{x_h} p(k), \text{ or } \int_{xl}^{xh} n(z)dz$$

Project III – Introduction and discussion

<u>The Project</u>

(1) Generate 10,000 Poisson events for <n>=100, determine the mean ($<n_{data}>$), and variance (σ^2) and, the standard error (σ) from this data sample. What is the error on $<n_{data}>$? What is the error on σ?

(2) From the data sample generated in (1), determine the confidence level for

$$n \geq <n> + \sigma, \; n \geq <n> + 2\sigma, \; \text{and } n \geq 120$$

(3) Generate 2 million Gaussian events with mean $\bar{x}$ =0 and σ=1.0, and determine from the data sample generated the confidence level for

$$x \geq k \times \sigma \quad \text{(k=0, 1, 2, 3, 4, 5, 6).}$$

A sample macro project3.C is available to illustrate the way to call ROOT random number generators.

Project III – Introduction and discussion

```
{
cout<<"random events are generated"<<endl;
gROOT->Reset();

Double_t Mean = 8.0, Sigma = 2.5;
Double_t ng =0.0, np=0.0;
Int_t    Nrun=10000, N=0, N18=0;

hr1  = new TH1F("hr1","Gsuaaian random data",400,-10,30);
hr2  = new TH1F("hr2","Poisson random data",400,-10,30);
hr3  = new TH1F("hr3","Poisson random data",400,-10,30);
hr1->GetXaxis()->SetTitle("n value");
hr2->GetXaxis()->SetTitle("n value");
hr3->GetXaxis()->SetTitle("n value");

for (Int_t i=0;i<Nrun;i++) {

   ng = gRandom->Gaus(Mean, Sigma);
   hr1->Fill(ng);
   np = gRandom->Poisson(ng);
   hr2->Fill(np);
   if(np>17.99){hr3->Fill(np);}
  }
}
```

Project III – Introduction and discussion

```
………

const Int_t kUPDATE = 500;
Float_t xrn, xbi, xps, xs1, xs2, xmain;

for ( Int_t i=0; i<10000; i++) {
  xrn   =   gRandom->Rndm(i);             // uniform Random Generator
  xbi   =   gRandom->Binomial(20,0.5);    //binomial distribution with N=20, p=0.5
  xps   =   gRandom->Poisson(10);         //Poisson distribution with <n>=10
  xmain =   gRandom->Gaus(-1,1.5);        // Gaussian with mean =-1.0, sigma=1.5
  xs2   =   gRandom->Landau(1,0.15);      // Landau distribution with center=1.0
                                             and Gamma=0.15
 }

…………………..
```

Project III – Introduction and discussion

Advanced Assignment

A Monte Carlo based, statistical experiment to determine the significance of an observation

Introduction

Discovery of the Top Quark Decay— The CDF experiment published its result on a search for the missing 6th quark in elementary physics. The experiment observed 15 candidates, an expected $5.96^{+0.45}_{-0.44}$ background/noise events.

Detection Method	N_{bkg} Expected Background	N_{SCand} Observed Signal Candidates	**Probability of Bkg Fluctuation**
di-lepton	$0.56^{+0.25}_{-0.13}$	2	0.12
Silicon vertex detector	3.1 ± 0.3	7	0.038
Soft lepton	2.3 ± 0.3	6	0.038
SUM	$5.96^{+0.45}_{-0.44}$	**15**	**0.0026**

The CDF collaboration has determined that the probability that the observed yield is consistent with the background is estimated to be 0.26%

Project III – Introduction and discussion

<u>The project</u>
Perform a Monte Carlo experiment to demonstrate that indeed the probability that the observed yield is consistent with the background is estimated to be 0.26%, using random number generators (Poisson, Gaussian and Uniform random number generators).

Hints:

(1) Go over the CDF paper (posted at the class site), and
(2) Calculate the ratio

$$\frac{N(N_{SCand} > 15)}{N(N_{SCand} = \text{any value})}$$

Summary of Lecture IV

Organization

- Project assignment
- Lecture Notes

What have been covered

- Max. Likelihood Fits, Errors
- Variable transformation
- Monte Carlo Simulation
- Examples
- Project III