D. Jason Koskinen
koskinen@nbi.ku.dk

Advanced Methods in Applied Statistics
Feb - Apr 2025

Photo by Howard Jackman

Additional Class Info

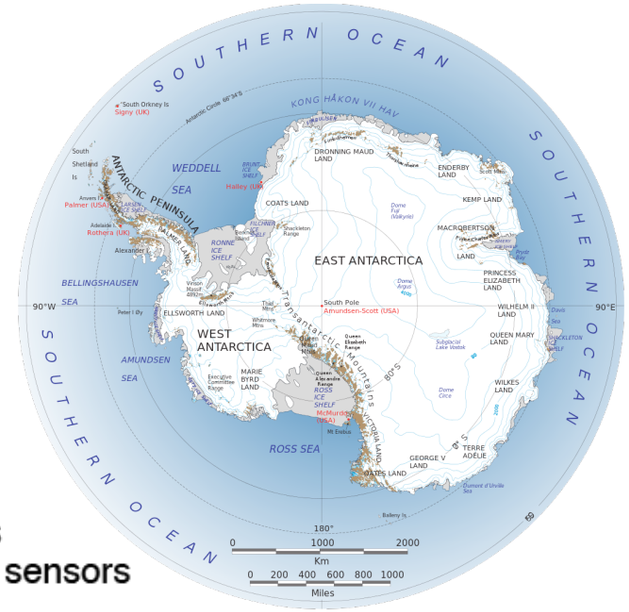
- No new lecture material in the afternoon, but there are slides about ‘Least Squares’ at the end of this presentation for those who are interested

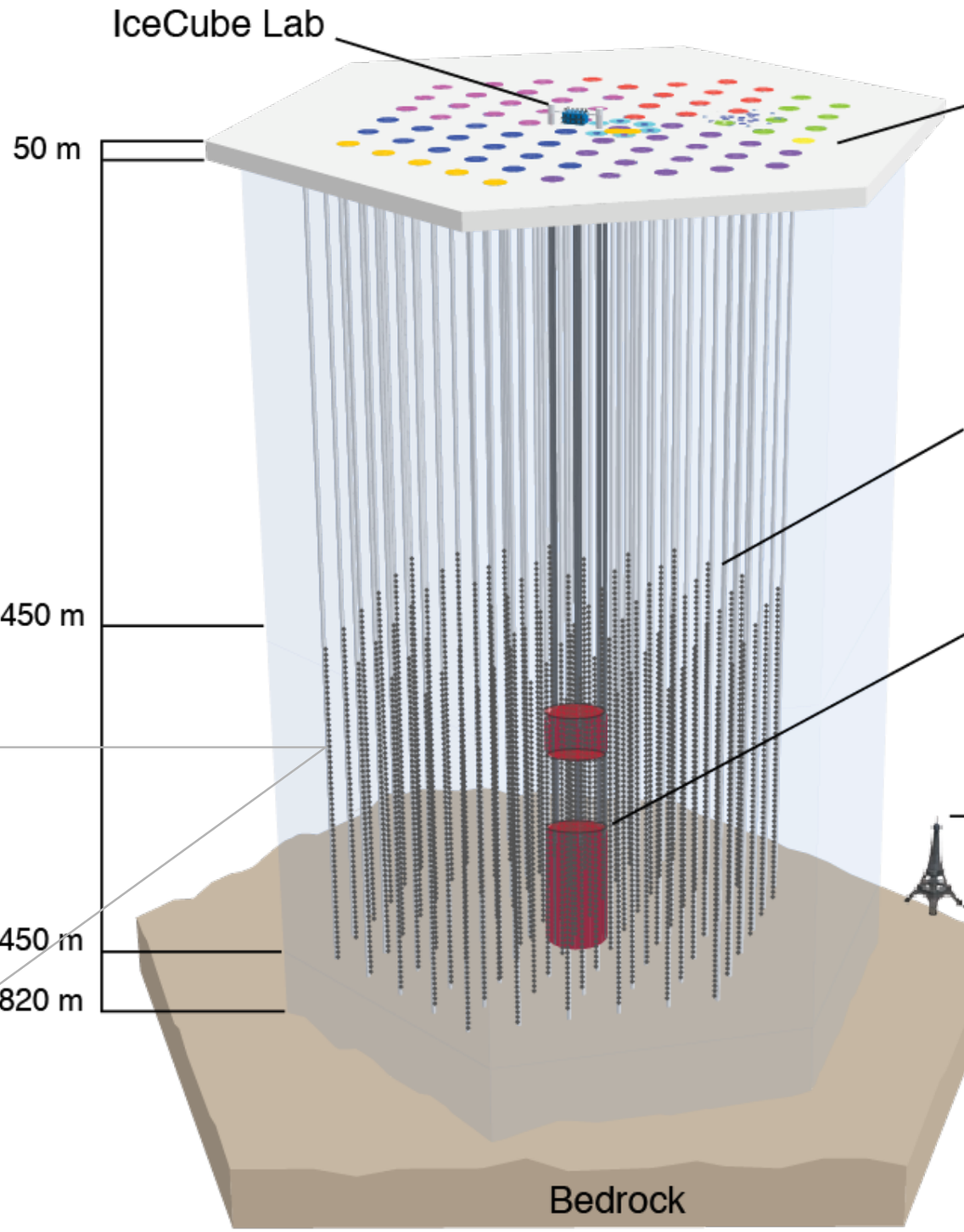
- This afternoon, there will be a discussion about the paper “Not Normal: the uncertainties of scientific measurements”
  - Paper can be (mostly) read over lunch; if you haven’t read it already

Monte Carlo (Simulation)

- Some things are complicated, too complicated for simple analytic tools. Both for physics processes as well as instrumental responses (noise, efficiency, thresholds, jitter, etc.)

IceCube



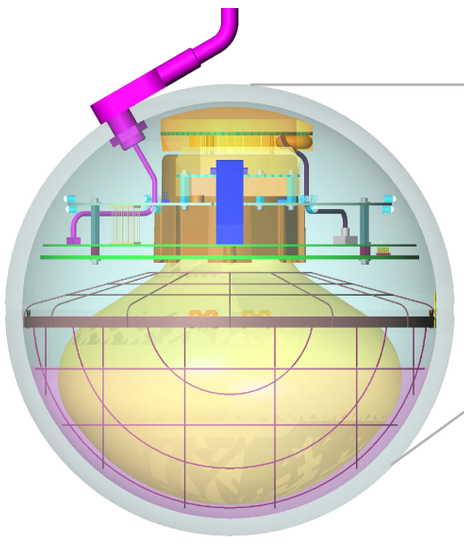


IceTop
81 Stations
324 optical sensors

IceCube Array
86 strings including
8 DeepCore strings
5160 optical sensors

DeepCore
8 strings-spacing optimized
for lower energies
480 optical sensors

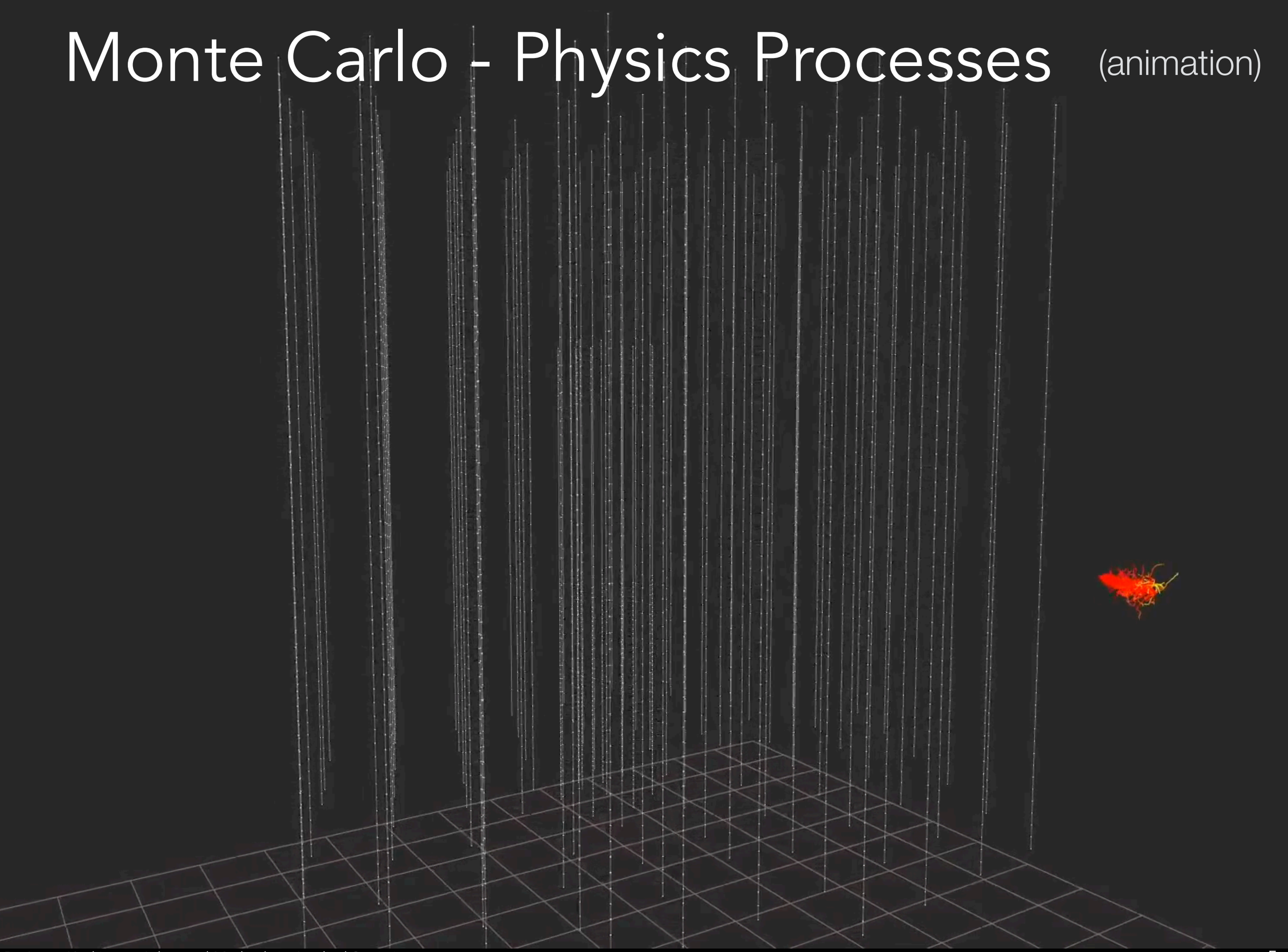
Eiffel Tower
324 m



IceCube Optical Sensor

~425 Scientists
14 Countries + Antarctica

Monte Carlo - Physics Processes (animation)



Monte Carlo - Physics Processes

- The previous movie is the simulation of a high energy muon moving through the IceCube detector at the South Pole.
- As the muon moves through ice, photons are emitted. The Cherenkov* photons are individually simulated as the thin strands (color denotes time since being emitted) as they scatter and then are ultimately absorbed.
- While the behavior that describes photon scatter and absorption may be simple, or complicated, it can be broken down at each photon ‘step’
  - Does the photon get absorbed; yes/no?
  - If ‘no’ absorption, then does it scatter; yes/no?
  - If ‘yes’ scatter, how much between 0-360°?
  - Move one step and repeat

If we can give a probability of each of these outcomes, then we can break the complex process into manageable pieces

- This is nearly impossible without computers

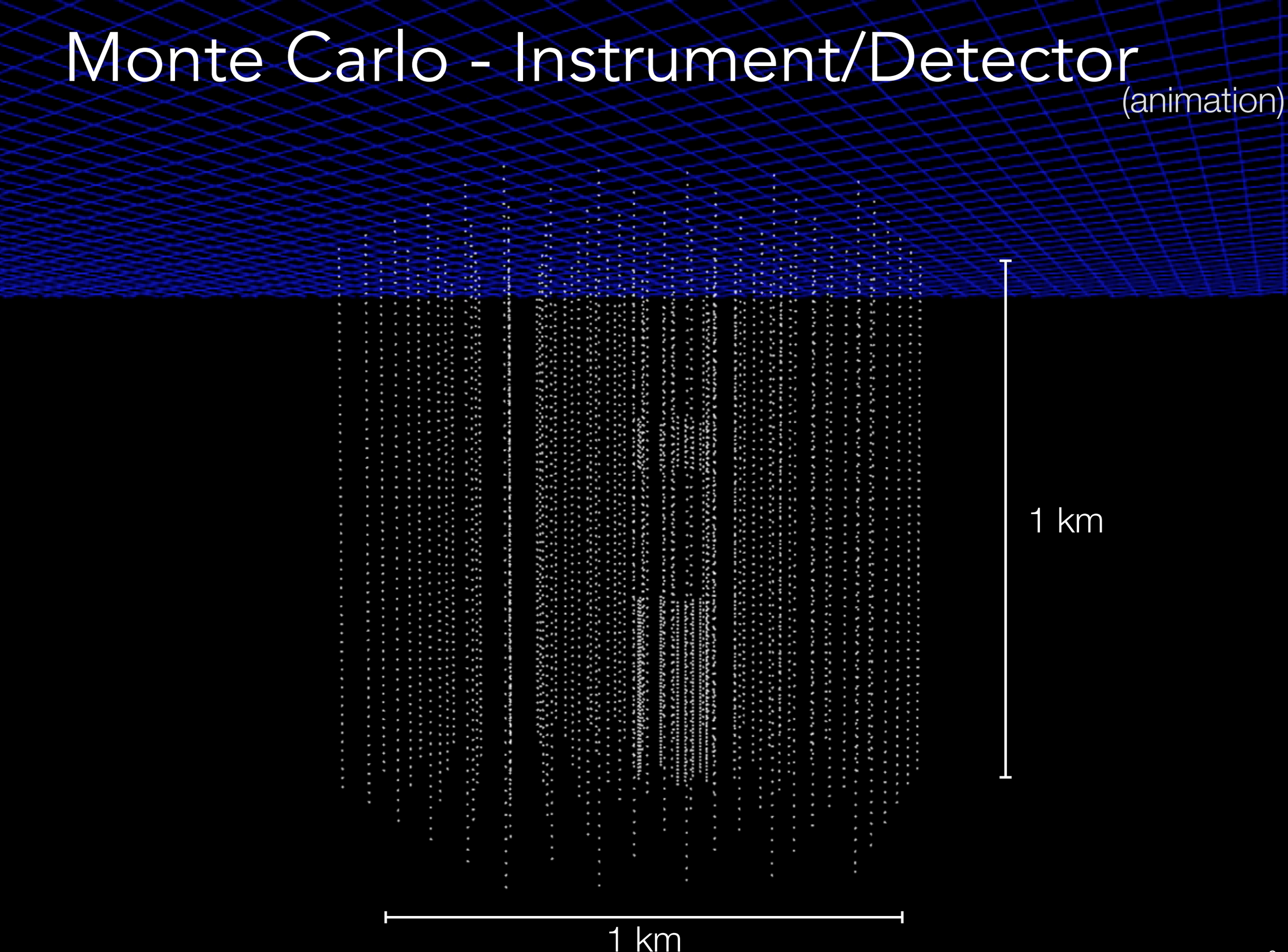
*https://en.wikipedia.org/wiki/Cherenkov_radiation

Monte Carlo - Instrument/Detector

- Even with a purely analytic treatment of the physics, detectors/telescopes/experiments/etc. are often complicated and extremely sensitive

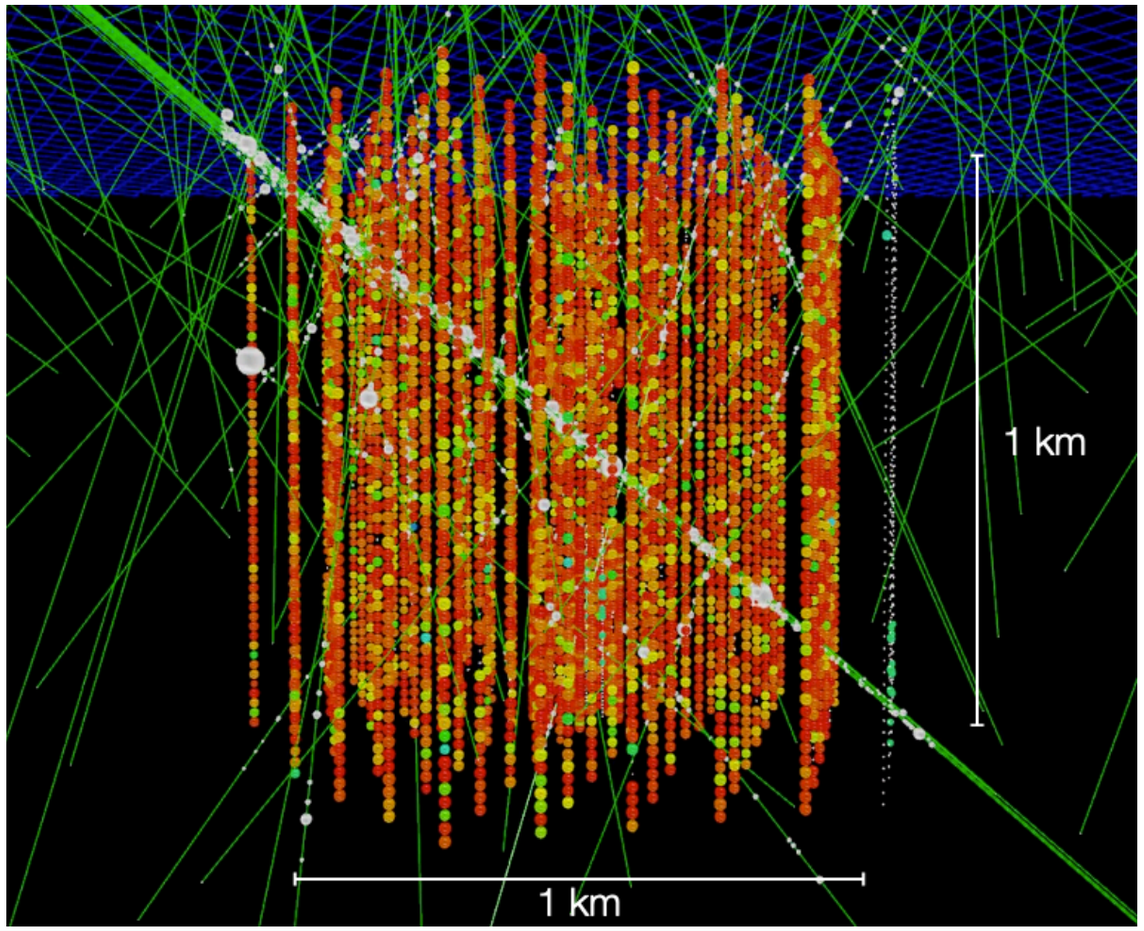
Monte Carlo - Instrument/Detector

(animation)



Monte Carlo - Instrument/Detector

- The previous movie is 10 ms of simulated data including noise and cosmic ray muons (green lines)
- The background rate (cosmic ray muons) is ~2200 Hz, whereas the neutrino signal for some analyses is 1 event every 1-3 months



Monte Carlo - Instrument/Detector

- The previous movie is 10 ms of simulated data including noise and cosmic ray muons (green lines)
- The background rate (cosmic ray muons) is ~2200 Hz, whereas the neutrino signal for some analyses is 1 event every 1-3 months
- Lots of Monte Carlo data makes sure you can optimize your analysis to keep signal and remove background, and has **many**, **many** more **benefits**
- It's all possible because of (pseudo) random number generators and computers

Random Numbers

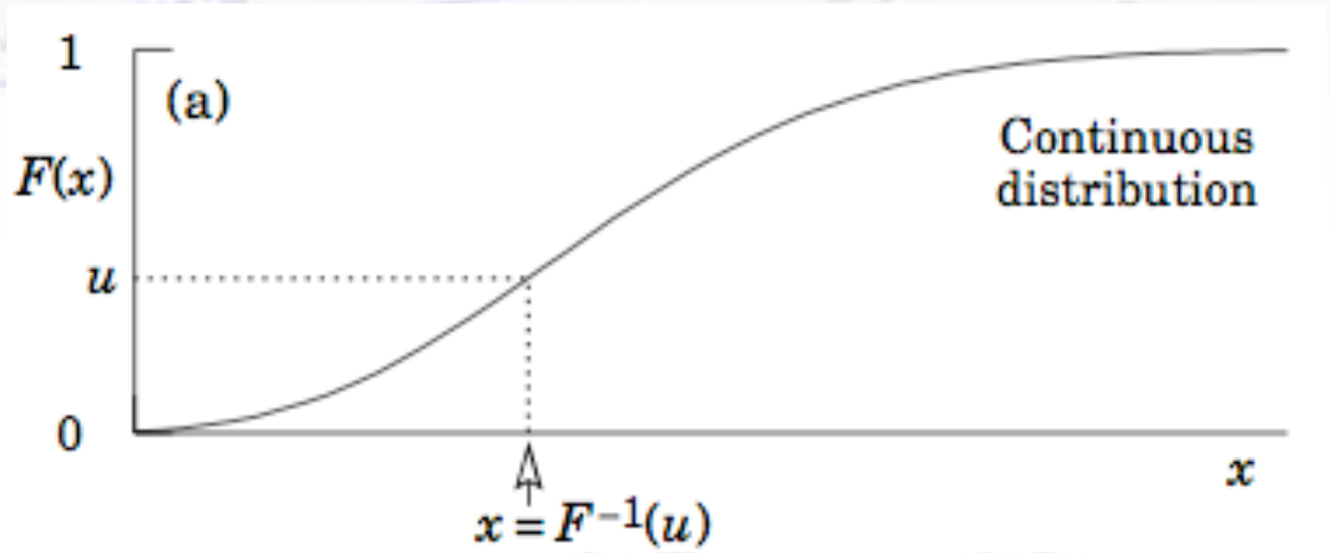
- A pillar of Monte Carlos and many analytic tools is the use of a generator that can produce ‘random’ values, most often in the range of 0 to 1
- Random in a linear fashion from 0-1 is nice because:
  - Probabilities go from 0-1
  - It is usually easy to map/transform linear in 0-1 to other ranges or functions. Note that zero can be problematic for some functions: 1/x, natural log (ln), etc.
  - Many default random number generators produce values that are linear in 0-1, e.g. numpy.random.uniform()
- There are two main ways to apply random numbers for Monte Carlo simulation and probability distribution function sampling: Transformation method and Accept-Reject

*T. Petersen, Applied Statistics

Transformation method

We have uniformly distributed random numbers r. We want random numbers x according to some distribution.

We want to try to find a function x(r), such that g(r) (uniformly distributed numbers) will be transformed into the desired distribution f(x).



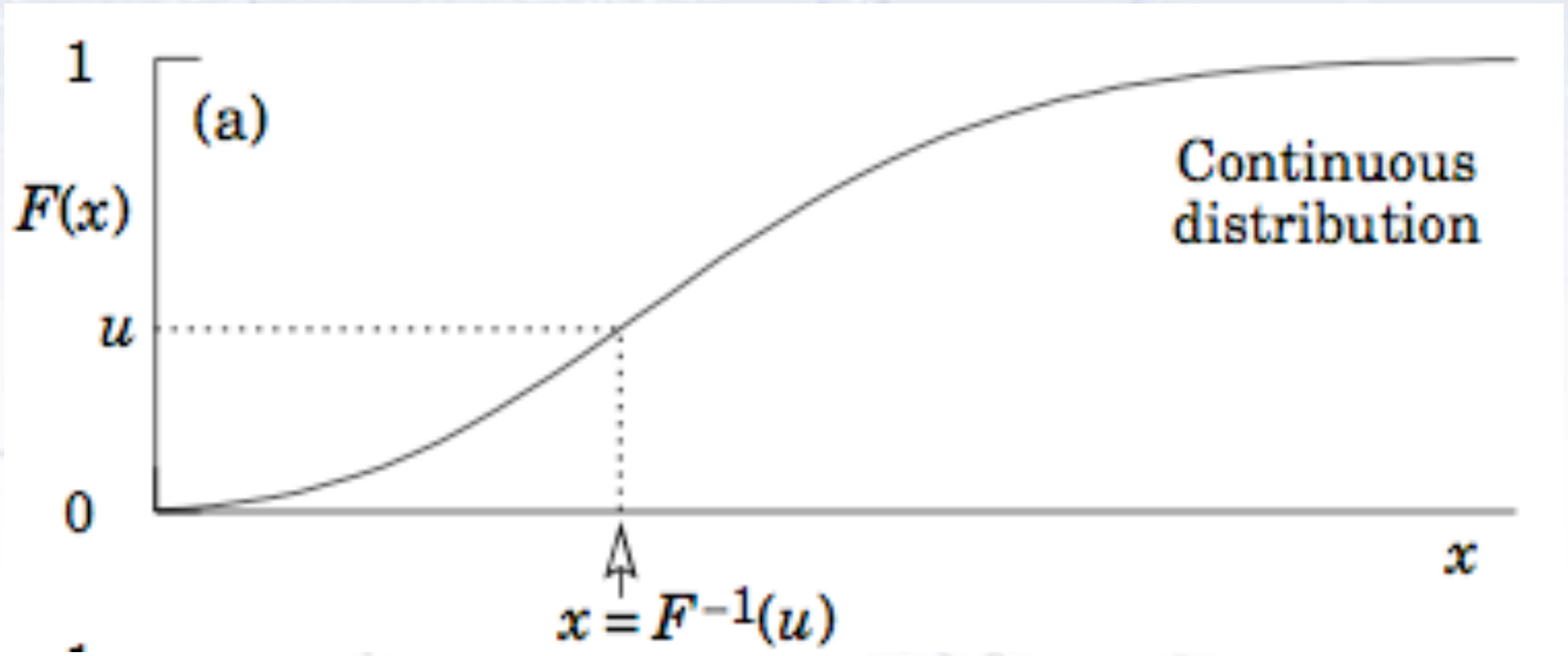
It turns out, that this is only possible, if one can (in this order):

Integrate f(x)

Invert F(x)

*T. Petersen, Applied Statistics

Transformation method



So the "recipe" can be summarised as follows:

- Ensure that the PDF is normalised!
- Integrate f(x) to get F(x) with the definite integral: $F(x) = \int_{-\infty}^{x} f(x')dx'$
- Invert F(x)

Now you can generate random numbers, x, according to f(x), by choosing $x = F^{-1}(u)$, where u is a random uniform number.

In Practice

- Transformation method is efficient when possible, but it is not always practical and will **NOT** be explicitly necessary in any problem set or exam for this course.
  - Except for gaussian, Poisson, and binomial distributions; but random number generation from 'common' distributions are available for all major coding languages
- Instead, we focus on the Accept-Reject method, which is straightforward and many of the inherent inefficiencies are rarely noticed because of modern computer power

Acceptance-Rejection Method

- For a probability distribution function that can nicely fit in a easily to integrate width/area/volume/etc. it is possible to generate a PDF-based random number generator

- So we need to be able to generate ‘random’ numbers

Random Number Generators

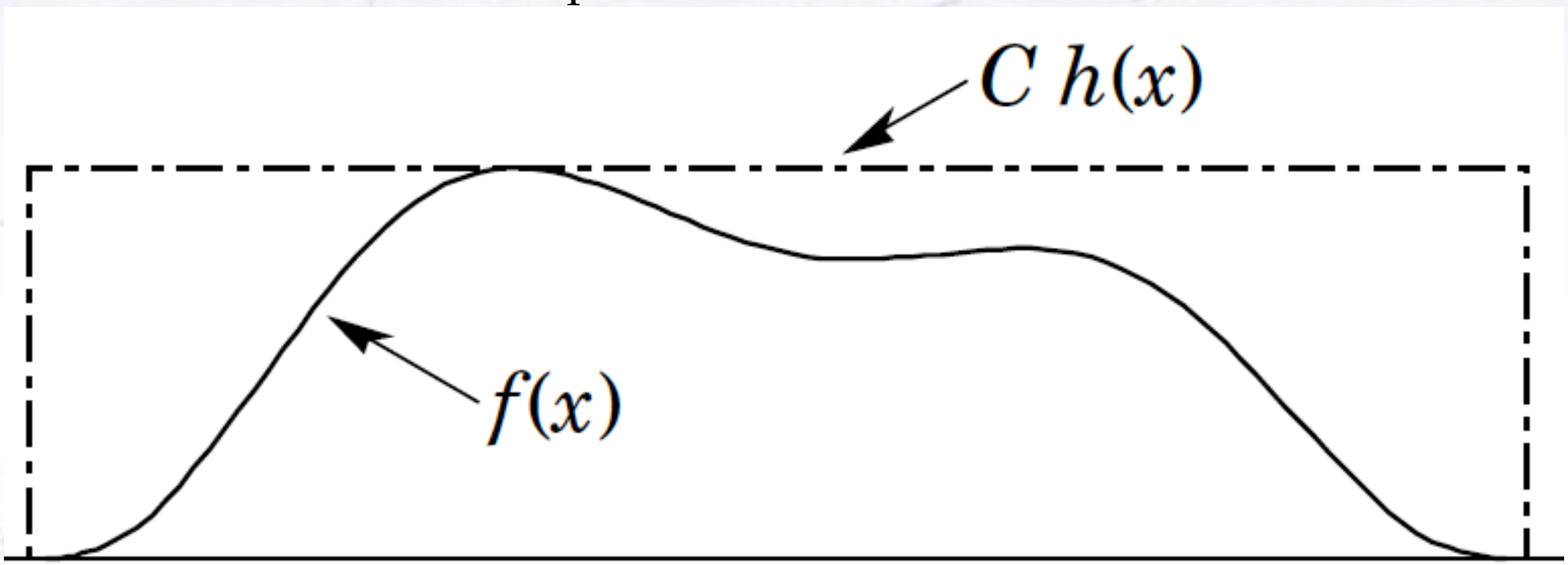
- Get your favorite random number generator (and I know that you all have favorites) and start randomly sampling
- Making plots is a good first step for any analysis. Try different plotting styles to literally see if you notice any regularity from your tested random number generator
    - Human eye/brain is good at pattern recognition amidst ‘noise’
    - See if there’s a sequence to the generator
    - Is there a ‘seed’ option? If so, does that change anything?
- Time the random number generator
    - Does it take 10 times longer to run if it produces 10 times more random numbers?

Accept-Reject method

(Von Neumann method)

If the PDF we wish to sample from is bounded both in x and y, then we can use the "Accept-Reject" method to select random numbers from it, as follows:

- Pick x and y uniformly within the range of the PDF.
- If y is below PDF(x), then accept x.



The main advantage of this method is its **simplicity**, and given modern computers, one does not care much about efficiency. However, it requires **boundaries**!

Precision

- Random numbers from generators are as accurate as the method, e.g. Mersenne twister, as well as the computational precision of the variable(s)
- Variable types related to int, float, and double have a characteristic precision, 8-bit, 32-bit, etc.

- For example, the float precision in some python versions is 53-bits, and therefore there is intrinsic rounding for precision at the scale of $1/2^{53}$

```
In [6]: 0.1 + 0.2
Out[6]: 0.30000000000000004
```

Take a look at The Perils of Floating Point for an interesting and brief discussion on computation numerical precision

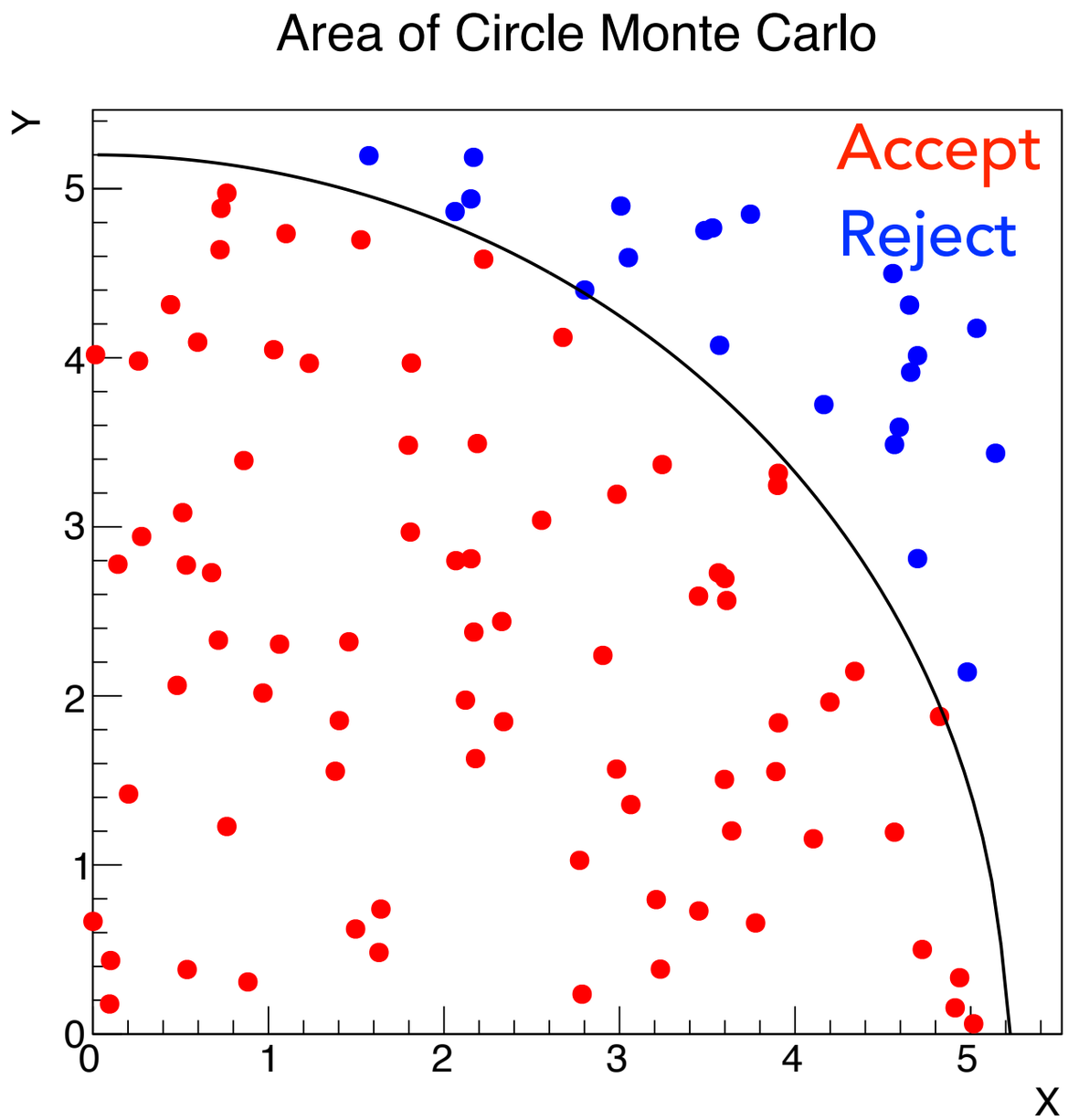
*ipython 4.0.1

Exercise 1: Classic & Simple Monte Carlo Usage

- Assume we do not know the value of π, but we want to calculate the area of a circle with only two things:
  - the equation $x^2+y^2=r^2$
  - random number generator
- Using a Monte Carlo technique, calculate the area of a circle with radius=5.2
- Show visualization, i.e. plot, of your method
- For the ambitious/creative: can you estimate the area using a Monte Carlo sampler in 1-dimension? I.e. not independently sampling x **and** y.

My Circle Area Visualization

- This is the classic illustration. Can you show your method in a different format that is understandable?

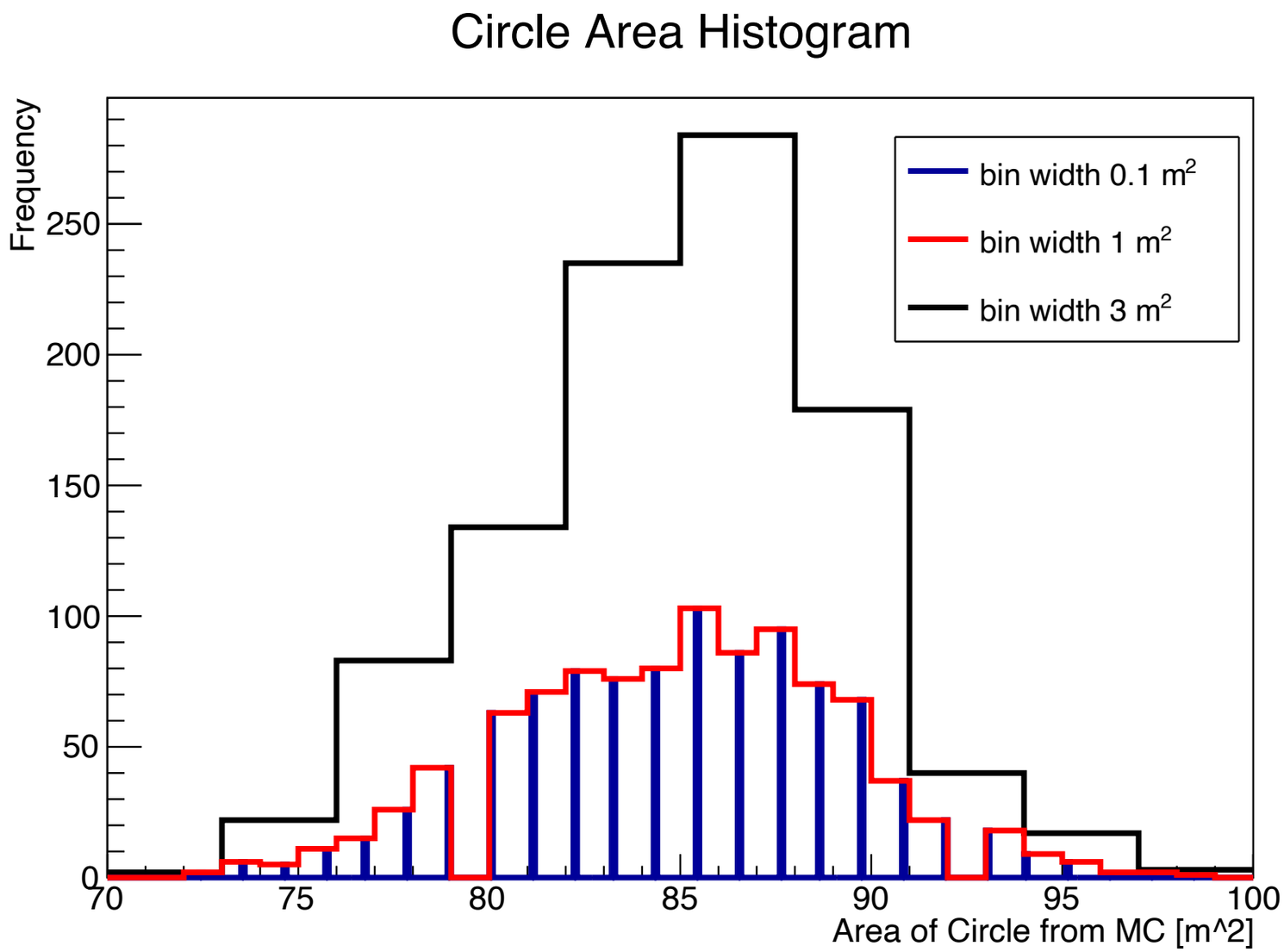


Exercise 2: Random Number Generator

- Using the previous code for estimating the area of a circle, plot the resulting area value for 1000 separate tests/trials using 100 throws per test for a radius of 5.2 meters
  - Each test/trial is a sample of 100 points resulting in a single estimate of the area of the circle.
- Does your previous code show actual randomness? How would you know?
- Make histograms of the frequency of the area for the **same** 1000 separate trials, i.e. not 3 separate histograms of 3 different 1000 trials
  - Using bin widths of 3 m^2, 1 m^2 , and 0.1 m^2
  - Are there gaps w/ no entries for certain values of the area? Should there be?

Dissecting a Plot

- I made 1,000 separate Monte Carlo trials, each having 100 random number generator throws to calculate the area of a circle with radius of 5.2 m

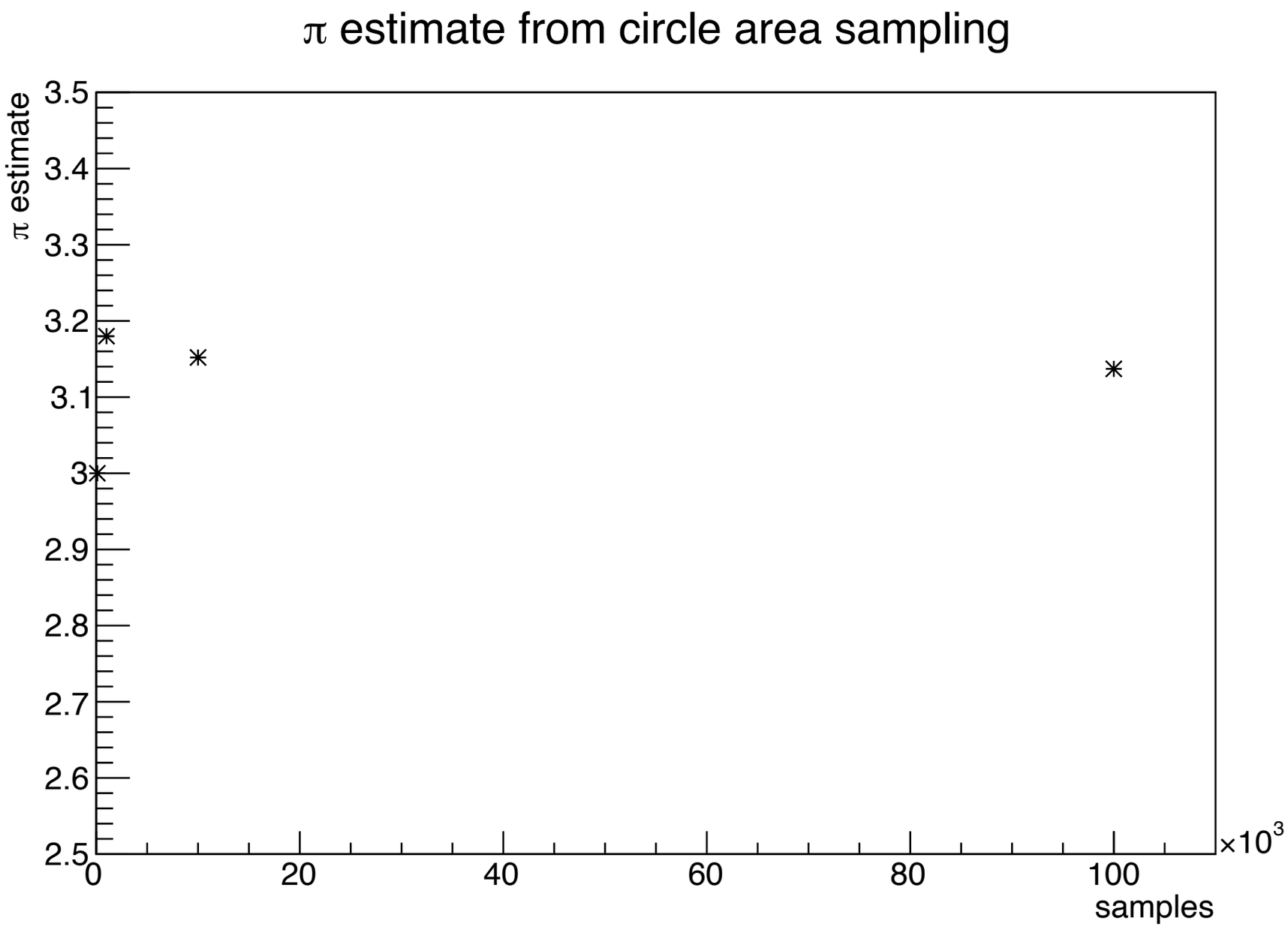


Exercise 3: Calculate the Precision of Pi

- Using your Monte Carlo and the circle equation $x^2+y^2=r^2$ find out the value of π knowing that the area is πr^2
  - After 10 successive 'throws' of your random number generator you have an estimate of the circle area. A bad estimate, but an estimate nonetheless. Using that estimate of the area, and knowing the radius (r=5.2), you can estimate π.
  - Repeat the estimation of π after 10 throws, 100 throws, 1000 throws, 10000 throws, and 100000 throws
  - Plot. On the y-axis have the estimate of π and on the x-axis have the number of throws.

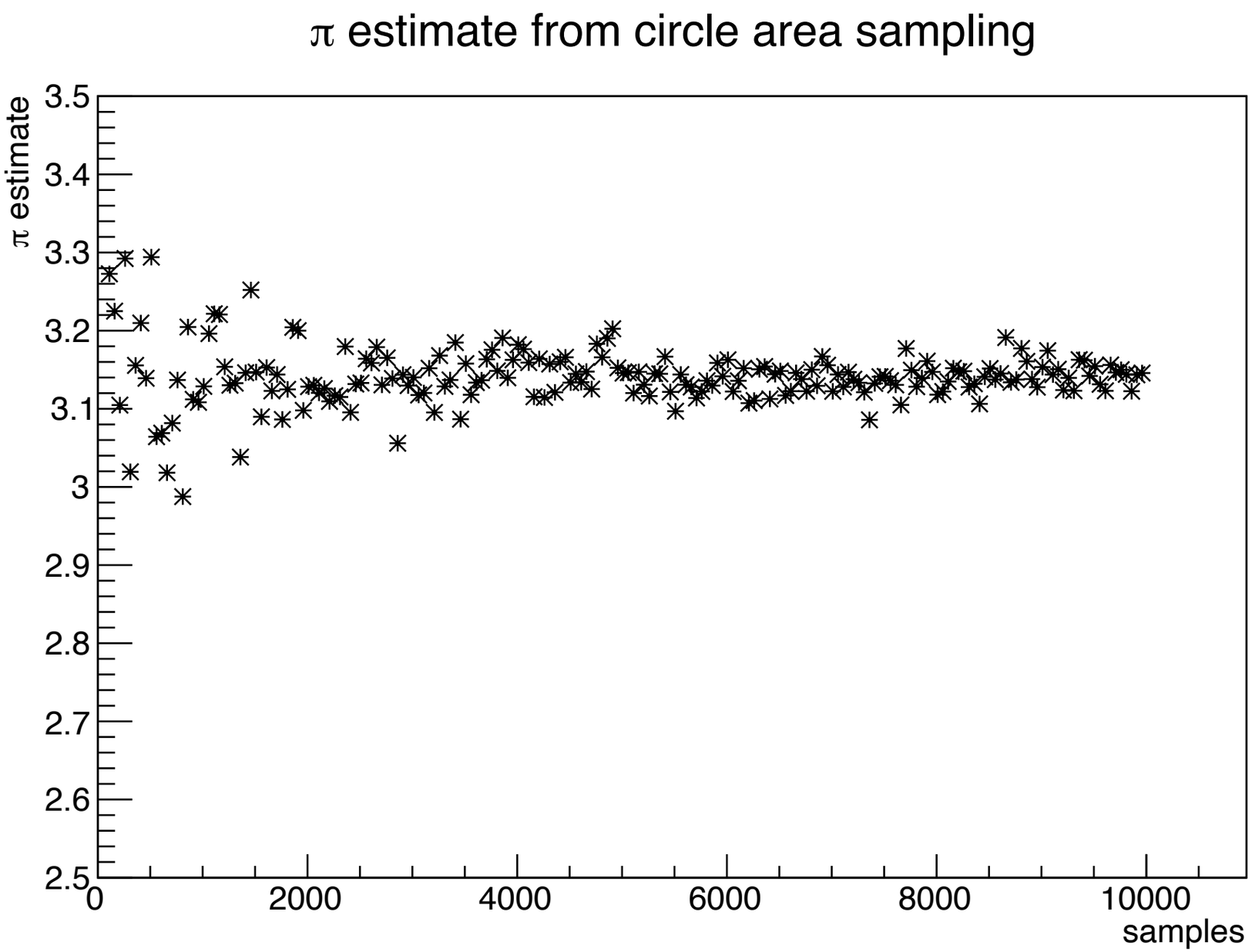
Calculate the Precision of Pi (1)

- Repeat the estimation of π after 10 throws, 100 throws, 1000 throws, 10000 throws, and 100000 throws



Calculate the Precision of Pi (2)

- Repeat the estimation of π for at least 100 different sampling points between 1-10000



For Next Week

- We will be using Monte Carlo techniques in our Likelihood fitting next week, which requires Monte Carlo sampling from different distributions. As such, **before class next week** you should have code that can:
  - Sample 100 values from a uniform probability distribution function over the range of $0 < x < 9.7$
  - Sample 103 values from a gaussian probability distribution function, centered at 0 and $\sigma^2 = 1.2$
    - Can use the inverse transform method*
    - Best option is to find a numerical software package
  - Plot the Monte Carlo generated data with histograms.
- Be able to generate multiple samples of Monte Carlo data and output/save the data sets to a common readable file format; either .csv or .txt

*https://en.wikipedia.org/wiki/Inverse_transform_sampling

Discussion

- Going to discuss article “Not Normal: the uncertainties of scientific measurements”
  - https://arxiv.org/abs/1612.00778
  - Look at some potential questions and talking points at Questions for the Not Normal paper discussion.docx

Extra

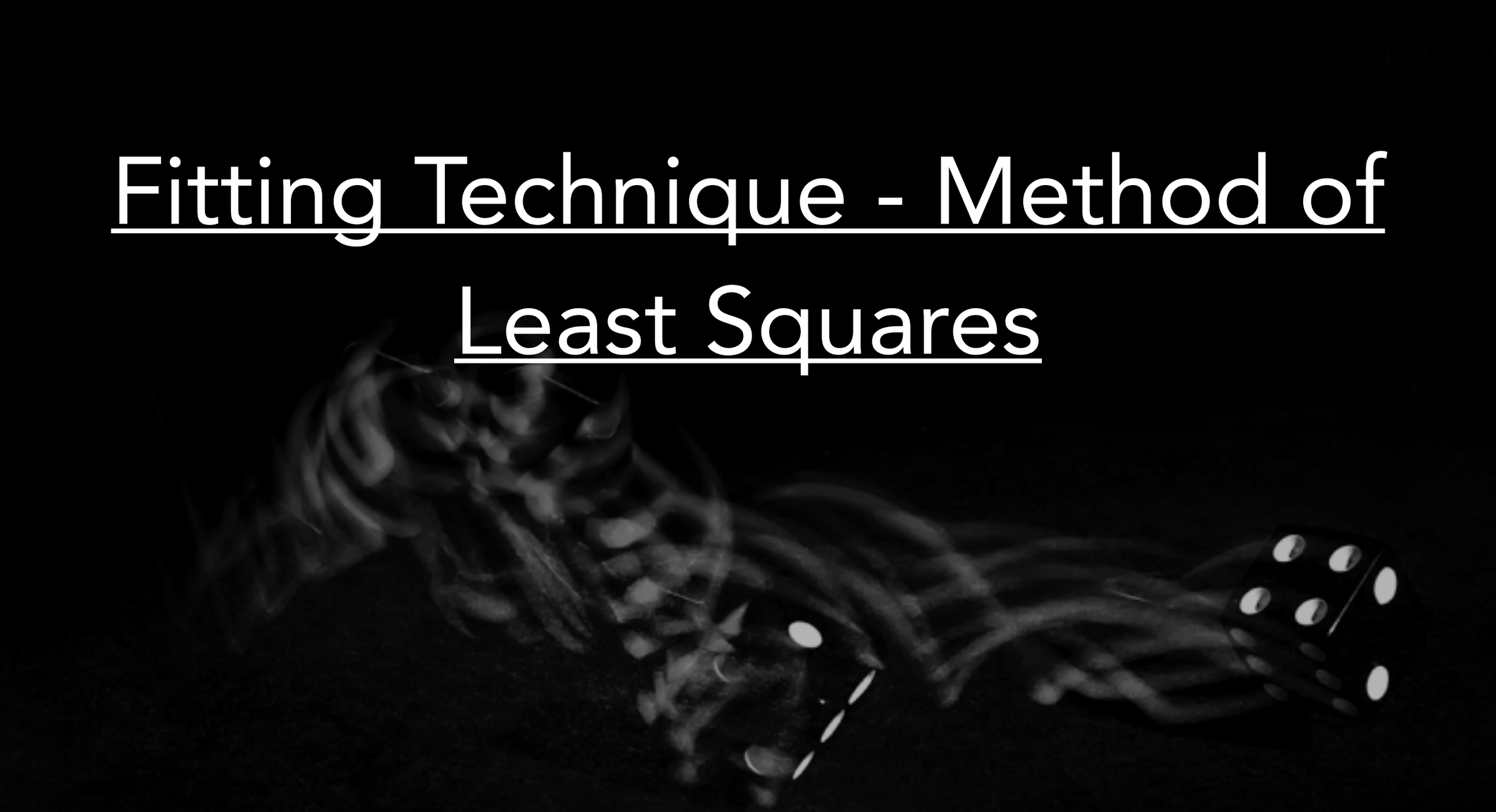
- In-depth look at Gaussian random number generation http://www.doc.ic.ac.uk/~wl/papers/07/csur07dt.pdf

Break -
Class discussion
restarts at 13:00

Note

- The following slides are included for an understanding of the Least Squares method(s). They will not be covered as a lecture, and the least squares method will not be part of any problem sets nor will there be a dedicated question on the exam which requires the use of a least squares fit.

Fitting Technique - Method of Least Squares



D. Jason Koskinen
koskinen@nbi.ku.dk

Advanced Methods in Applied Statistics

Photo by Howard Jackman

Method of Least Squares

- In today's lecture:
  - Introduction
  - Linear Least Squares Fit
  - Least Squares method estimate of variance
  - Non-linear Least Squares
  - Least Squares as goodness-of-fit statistic
  - Least Squares on binned data
- A lot, lot more math and analytic coverage than usual in the following slides. Should be used as reference material, but focus on **using** your least squares minimization routines.

Material from D. Grant derived entirely from lectures by A. Bellerive, G. Cowan, and D. Karlen

Method of Least Squares

- Introduction
  - Most frequently used fitting method, but has no general optimal properties that would make that the case.
  - When the parameter dependence is linear, the method produces unbiased estimators of minimum variance.
  - The method is applied as follows:
    - for observation points (x) experimental values are measured (y). The true functional form is defined by L parameters:

$$f_i = f_i(\theta_1, ..., \theta_L)$$

    - To find parameter estimates, θ, we minimize:

$$X^2 = \sum_i w_i (y_i - f_i)^2$$

weight expressing accuracy of y

Method of Least Squares

- Introduction
  - The method is applied as follows (cont.):
    - In the case of constant accuracy, all w = 1.
    - If the accuracy for y is given by σ_i then $w_i = 1/\sigma_i^2$
    - If the observations are correlated, then the minimization function becomes:

$$X^2 = \sum_{i=1}^{N}\sum_{j=1}^{N}(y_i - f_i)V_{ij}^{-1}(y_j - f_j)$$

V_{ij} is the covariance matrix

  - where the values for independent variable(s) (generally x) are assumed to have be known precisely, i.e. no uncertainties.

Method of Least Squares

- Introduction
  - The method is applied as follows (cont.):
    - In many cases the measured value (y) can be regarded as a Gaussian random variable centered about the true value, as expected from the Central Limit Theorem as long as the total error is the sum of a large number of small contributions.
    - For a set of N independent Gaussian random variables (y_i) of unknown mean (λ_i) and different, but known, variance (σ_i^2) then the joint PDF can be written:

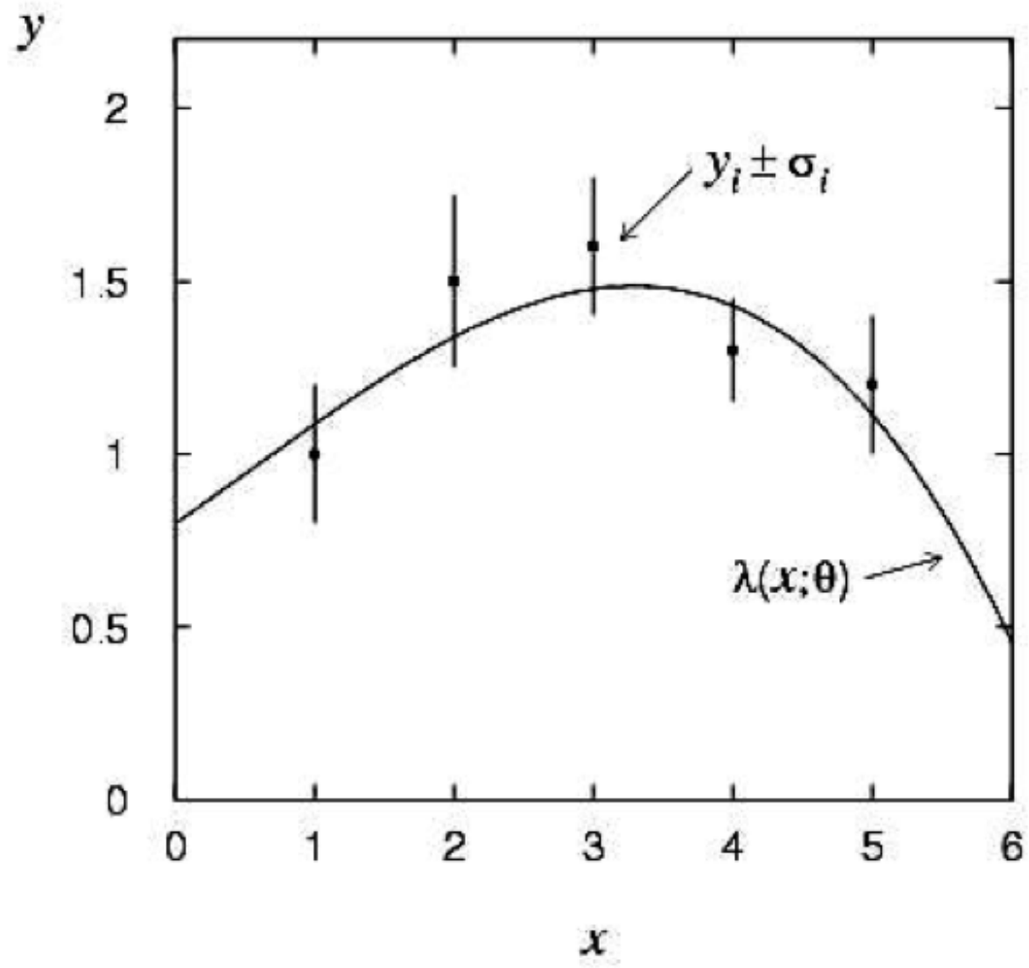
$$g(\vec{y};\vec{\lambda},\vec{\sigma}^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}(\frac{y_i-\lambda_i}{\sigma_i})^2}$$

Method of Least Squares

- The method is applied as follows (cont.):
    - Again, the measurements are related to x, which is known precisely. We can write the true value in terms of a function of x with unknown parameters θ:

$$\lambda = \lambda(x; \vec{\theta})$$

    - The goal is to estimate the parameters (θ) with the least squares method; a simple evaluation of the goodness of fit of the hypothesized function above.



Method of Least Squares

- Introduction
  - The method is applied as follows (cont.):
    - The likelihood function is given by:

$$L(\vec{\theta}) = L(x;\vec{\theta}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}(\frac{y_i - \lambda_i(x;\vec{\theta})}{\sigma_i})^2}$$

- which corresponds to the log-likelihood function:

$$\ln L(\vec{\theta}) = -\frac{1}{2} n \ln 2\pi + \ln(\prod_{i=1}^{N} \sigma_i^{-1}) - \frac{1}{2}\sum_{i=1}^{N}(\frac{y_i - \lambda_i(x;\vec{\theta})}{\sigma_i})^2$$

$$\ln L(\vec{\theta}) = const - \frac{1}{2}\sum_{i=1}^{N}(\frac{y_i - \lambda(x_i;\vec{\theta})}{\sigma_i})^2$$

$$-2\ln L(\vec{\theta}) = \sum_{i=1}^{N}(\frac{y_i - \lambda(x_i;\vec{\theta})}{\sigma_i})^2 + const$$

Method of Least Squares

- Introduction
  - The method is applied as follows (cont.):
    - One may maximize lnL, or minimize:

$$\chi^2(\vec{\theta}) \equiv \sum_{i=1}^{N} \left(\frac{y_i - \lambda(x_i;\vec{\theta})}{\sigma_i}\right)^2$$

    - The errors on the estimated parameters are obtained by evaluating the corresponding one standard deviation departure from the least-squares estimate:

$$\chi^2(\vec{\theta}) = \chi^2(\tilde{\vec{\theta}}) + 1 \qquad \text{since} \qquad \chi^2(\vec{\theta}) = -2\ln L(\vec{\theta}) + const$$

    - Thus, if the **measurements** are **Gaussian** distributed, then the least square method can be equivalent to the maximum likelihood method. Further, the observables will be linear functions of the parameters and follow a chi-square distribution.

Method of Least Squares

- Linear LS Fit
  - If the observables are linear functions of the unknown parameters and the weights are independent of the parameters, then the LS method has an exact solution that can be written in a closed form.
  - Consider

$$\lambda(x;\vec{\theta}) = \sum_{j=1}^{N} a_j(x)\theta_j$$

  - the a(x) terms are any linearly independent function of x such that λ is linear in the parameters θ. The a(x) are generally not linear in x, but are linearly independent of each other.
  - In this case, an analytic solution for the estimators and their variances exists. The estimators will be unbiased from the minimum variance bound condition regardless of the number of measurements and the PDF of the individual measurements.

Method of Least Squares

- LS Fit for a polynomial
  - As a hypothesis for $\lambda(x;\vec{\theta})$, you might want to use a polynomial of order m, in the case of m+1 parameters, e.g.

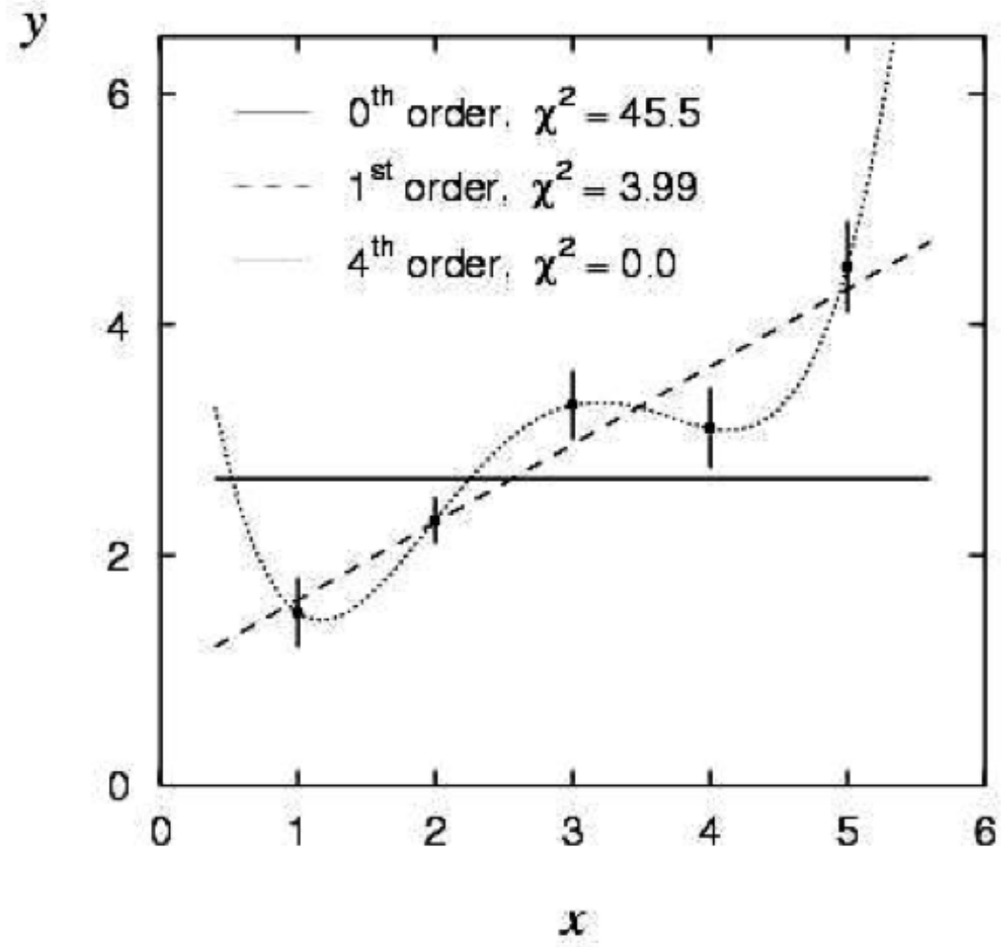
$$\lambda(x;\vec{\theta}) = \sum_{j=0}^{m} x^j \theta_j$$

  - This is a special case of the linear LS method with linearly independent weights:

$$a_j(x) = x^j$$

  - Thus, just as before,

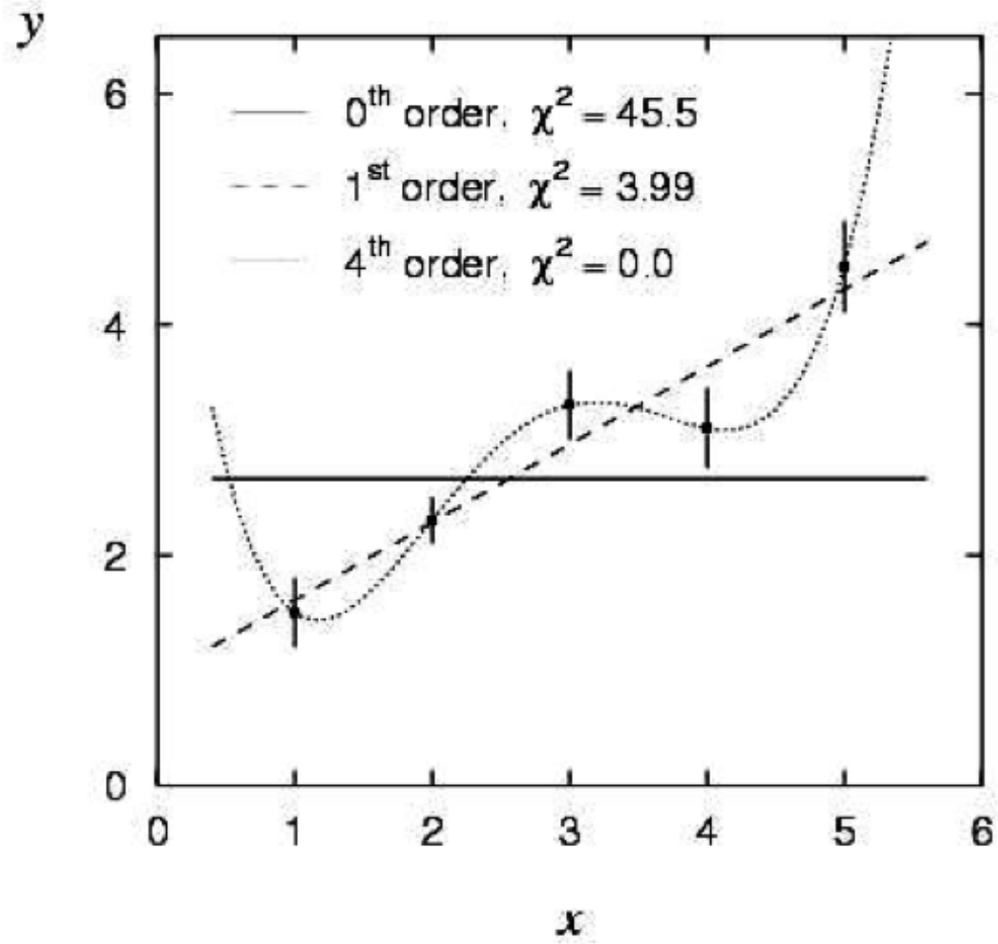
$$\chi^2 = (\vec{y} - A\vec{\theta})^T V_{ij}^{-1} (\vec{y} - A\vec{\theta})$$



Method of Least Squares

- LS Fit for a polynomial
  - The illustration to the right is for different polynomial fits which are possible for least squares, including a flat constant, i.e. 0th order polynomial
  - With the following data, test a similar least squares fit for the points defined by: x = (0.0, 1.0, 2.0, 3.0, 4.0, 5.0) and y = (0.0, 0.8, 0.9, 0.1, -0.8, -1.0) at different polynomial orders
- Similar to the illustration, calculate the chi-square assuming each point has a uncertainty in y of ±0.5

*note the x and y data in the text is not the same as what is shown in the plot



Least Squares Examination

- Using your random number generator, sample from a gaussian distribution of your own choosing, i.e. width and mean, for n=10, 100, 1000, and 10000 throws and fit each with a 2nd and 3rd order polynomial least squares fit. Use histograms.
- Assume any negative predictions from the resulting polynomial fit are zero.
- Calculate the chi-square for each combination of trials and polynomial least squares fits
- The uncertainty is related to the expected poisson fluctuations from your samples of the random number generator.
- Plot the resultant fits and see what happens for higher order polynomial fits, e.g. order 5, 7, 8, 12, whatever, etc… as the number of data points (random number generator throws) increases

Least Squares Examination (cont.)

- From the resultant fits for higher order polynomial fits, e.g. order 5, 7, 8, 12, whatever, etc… as the number of data points (random number generator throws) increases, calculate the chi-square using the uncertainty (or weight) as the expected poisson fluctuation
  - Where do the polynomial fits give a good ‘fit’ to a gaussian, even though a gaussian distribution and polynomial are not the same?
  - How does the agreement change as a function of polynomial order or throws from the random number generator?

Extra

Examples of Least Squares Routines

- Non-Linear LS Fit
    - Need a numerical method to evaluate the estimators and their covariance matrix. ROOT and other packages provides this capability to fit any type of function, including that provided by a user defined routine.
    - Examples of user routine for a chi-square numerical minimization:
        - Polynomial of order m:

$$y(t) = x_0 + x_1 t + x_2 t^2 + ... + x_m t^m$$

        - Gaussian:

$$y(t) = x_1 e^{-\frac{1}{2}\left(\frac{t-x_2}{x_3}\right)^2}$$

$$x_1 = \text{amplitude}$$

$$x_2 = \text{mean}$$

$$x_3 = \text{standard deviation}$$

Examples of Least Squares Routines

- Non-Linear LS Fit
  - Examples of user routine for a chi-square numerical minimization:
    - Exponential:

$$y(t) = x_1 e^{-x_2 t}$$

    - Trigonometric:

$$y(t) = x_1 sin(x_2 t) \qquad y(t) = x_1 cos(x_2 t)$$

$$y(t) = x_1 sin(x_2 t + x3) \qquad y(t) = x_1 cos(x_2 t + x3)$$

  - Damped Oscillator

$$y(t) = x_1 e^{-x_2 t} cos(x_3 t + x4)$$

Examples of Least Squares Routines

- Non-Linear LS Fit
  - Examples of user routine for a chi-square numerical minimization:
    - Breit-Wigner:

$$y(t) = \frac{2}{\pi x_2} \frac{x_3 x_2^2}{4(t - x_1)^2 + x_2^2}$$

$$x_3 = \text{amplitude}$$

$$x_1 = \text{mean}$$

$$x_2 = \text{width}$$

- We of course want to find the relation between true values, according to some hypothesis, and measured quantities, y, at known observations with no errors, x. e.g.: $f_j(\vec{y}; \lambda) = y_j - h_j$

Method of Least Squares

- Non-Linear LS Fit
  - We need to find the minimum function:

$$\chi^2 = (\vec{y} - \vec{h}(x))^T G_y (\vec{y} - \vec{h}(x))$$

  - The convergence of a numerical (iterative) procedure will depend on if we are in an area of the phase space where the chi-square function is similar to a quadratic form. That is to say, the non-linear case first approximation is the starting point for the numerical procedure. For this simple algorithm to work we must be near the absolute minimum.

  - We expand the function around a set of first approximations for the (r) unknowns or parameters:

$$f_j = y_j - h_j \rightarrow \phi$$

$$f_j(x)_{true} = f_j(x_0) + (\frac{\partial f_i}{\partial x_i})_{\vec{x_0}}(x_i - x_{i0}) + ... = 0$$

estimates

$$\sum_{i=1}^{r}$$

Method of Least Squares

- Non-Linear LS Fit
  - We expand this about

$$\vec{x}_0 \equiv \vec{\theta}_0 = (\theta_1, ..., \theta_r)$$

$$f(y_j; \vec{\theta}) = f(y_j; \vec{\theta}_0) + \sum_{i=1}^{r} \left(\frac{\partial f_i}{\partial x_i} \right)_{\vec{\theta}_0} (x_i - \theta_i)$$

  - Which gives us:

$$\chi^2 = (\vec{c} + A\vec{\xi})^T G_y (\vec{c} + A\vec{\xi})$$

$$\vec{\xi} \equiv \vec{x} - \vec{\theta}$$

$$\vec{c}_j = f_j(\vec{y}; \vec{\theta}_0) = y_j - h_j(\vec{\theta}_0)$$

  - elements of A:

$$a_{jl} = \left(\frac{\partial f_j}{\partial x_l} \right)_{\vec{\theta}_0} = - \left(\frac{\partial h_j}{\partial x_l} \right)_{\vec{\theta}_0}$$

  - and G is the inverse variance matrix.

Method of Least Squares

- Non-Linear LS Fit
  - Note that the second derivative must of course be positive at the minimum when the chi-square is of quadratic form:

$$\frac{1}{2}\left(\frac{\partial^2\chi^2}{\partial x_1 \partial x_2}\right)_{\vec{x}=\tilde{\theta}} = A^T G_y A > 0$$

  - A step beyond the simple iterative method, known as step-size reduction, applies the fact that on each side of the minimum the first derivative changes sign and the second derivative is positive.

Method of Least Squares

- LS Fit as a goodness-of-fit
    - The value of the chi-square minimum reflects the agreement between data and hypothesis and can thus be used as a goodness-of-fit test statistic:

$$\chi^2_{min} = \sum_{i=1}^{N} \frac{(y_i - \lambda(x_i;\hat{\theta}))^2}{\sigma_i^2}$$

    - where our hypothesized function form is given by λ.
    - If the hypothesis is correct, then the test statistic, t, follows the chi-square pdf:

$$f(t;n_d) = \frac{1}{2^{n_d/2}\Gamma(n_d/2)} t^{n_d/2-1} e^{-t/2}$$

    - where n_d is the number of data points - number of fitted parameters.

Method of Least Squares

- LS Fit as a goodness-of-fit
    - The chi-square pdf has an expectation value equal to the number of degrees of freedom such that if the minimum chi-square is approximately the number of degrees of freedom then the fit is considered “good.”
    - We can find the p-value, here the probability of obtaining a chi-square minimum as large as the one measured, or higher, if the hypothesis is correct:

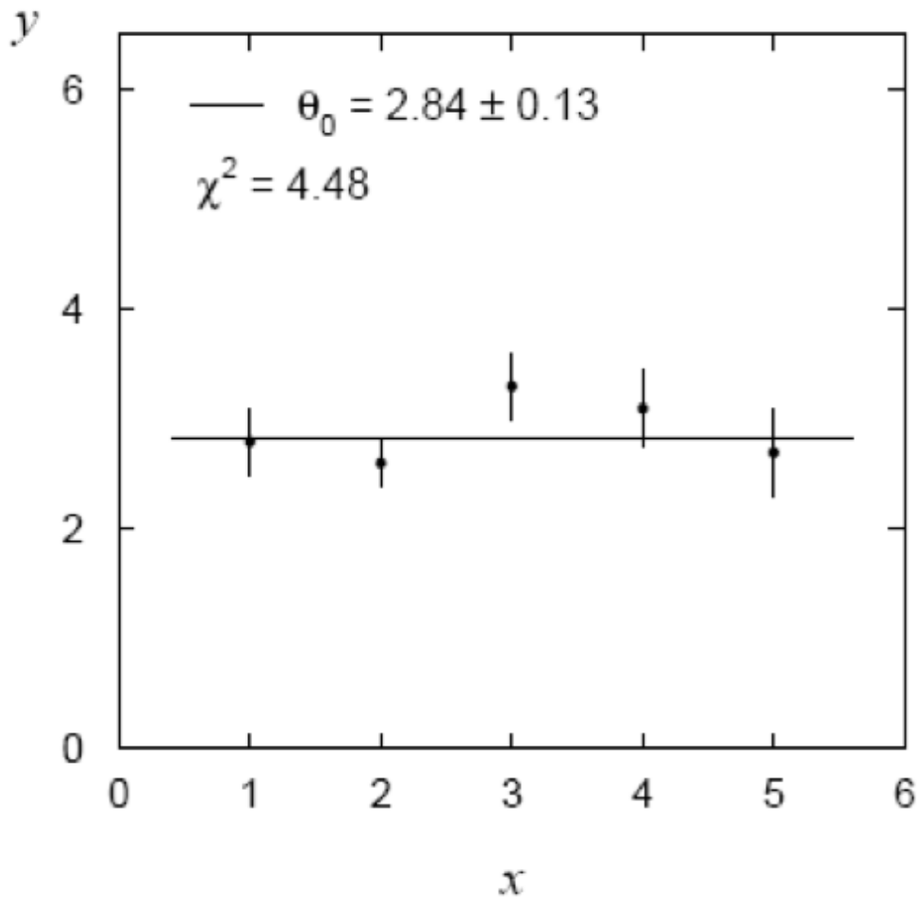
$$p = \int_{\chi^2_{min}}^{\infty} f(t; n_d) dt$$

    - From the polynomial fit example:
        - 2 parameter fit: $\chi^2_{min} = 3.99 \qquad n_d = 5 - 2 = 3 \qquad p = 0.263$
        - 1 parameter fit: $\chi^2_{min} = 45.5 \qquad n_d = 5 - 1 = 4 \qquad p = 3.1 \times 10^{-9}$

*results from illustration in slide 36

Method of Least Squares

- LS Fit as a goodness-of-fit vs. statistical error
  - It is important to note that a small st
    fit, nor does a good fit imply small s
  - The curvature of the chi-square near
    statistical errors
  - The value of the chi-square minimur
  - For horizontal line fit, move the data
    errors on the points the same.



Variance is same as previously, so the chi-square minimum is now "good"

Method of Least Squares

- LS Fit as a goodness-of-fit vs. statistical errors
    - The variance of the estimator (statistical error) tells us that if the experiment were repeated many times, the width of the distribution of the estimates, but not if the hypothesis, is correct.
    - The p-value tells us that if the hypothesis is correct, and the experiment repeated many times, what fraction of those will give equal or worse agreement between data and hypothesis according to the chi-square minimum test statistic.
    - Thus, a low p-value may indicate the hypothesis is wrong, due to systematic error.

Method of Least Squares

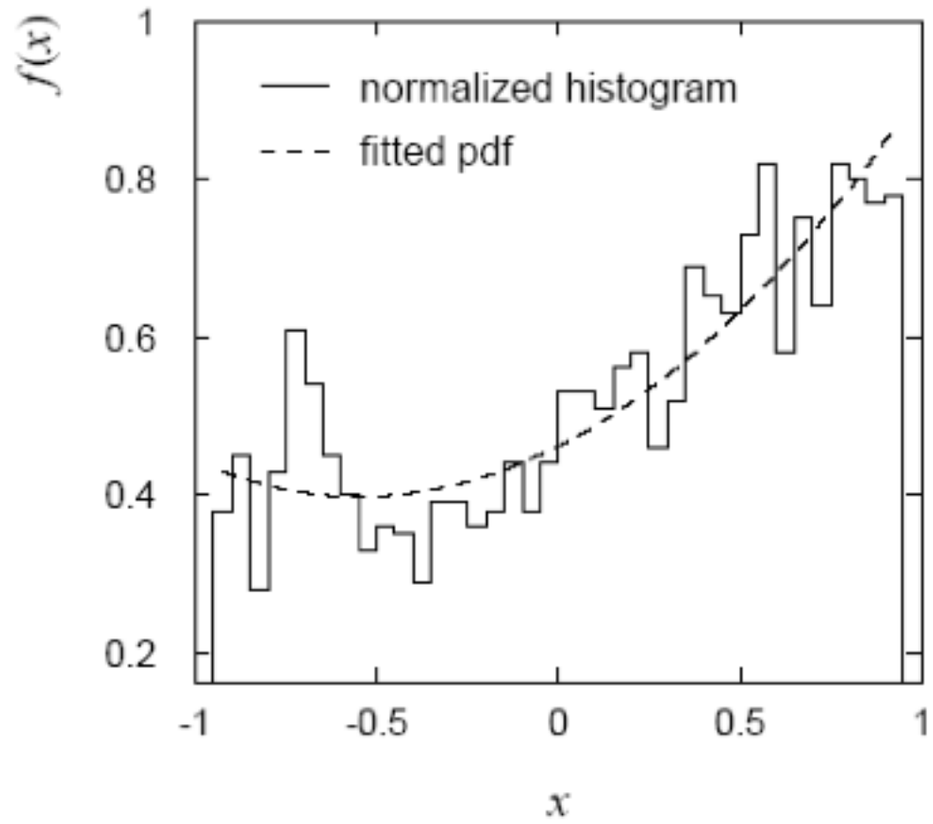
- LS method with binned data
  - If the data is split into N bins, with bin i containing n_i entries, there is a probability for an event to populate, p_i, that bin. Our hypothesized pdf is:

$$f(x;\vec{\theta})$$

  - The expected number of events in each bin is given by:

$$\lambda_i(\vec{\theta}) = n\int_{x_i^{min}}^{x_i^{max}} f(x;\vec{\theta})dx = np_i(\vec{\theta})$$

$$n = \sum_{i=1}^{N} n_i$$



Method of Least Squares

- LS method with binned data
    - Now for the fit we minimize

$$\chi^2(\vec{\theta}) = \sum_{i=1}^{N} \frac{(y_i - \lambda_i(\vec{\theta}))^2}{\sigma_i^2}$$

    - where our variances are not known a priori. We treat the y terms as Poisson random variables and, in place of the true variance, take either:

$$\sigma_i^2 = \lambda_i(\vec{\theta}) \text{ (Least Square method)}$$

$$\sigma_i^2 = y_i \text{ (Modified Least Square method)}$$

    - Note that the modified least squares is sometimes easier to compute, but the chi-square minimum statistic no longer follows the chi-square pdf if some of the bins have few or no entries.

Method of Least Squares

- LS method with binned data
    - We lose a degree of freedom because of the normalization condition:

$$\sum n_i = n$$

    - such that the chi-square minimum statistic will follow:

$$f(\chi^2, N-1-L)$$

    - assuming the model consists of L independent parameters.
    - It is NOT correct to fit for the normalization, e.g.

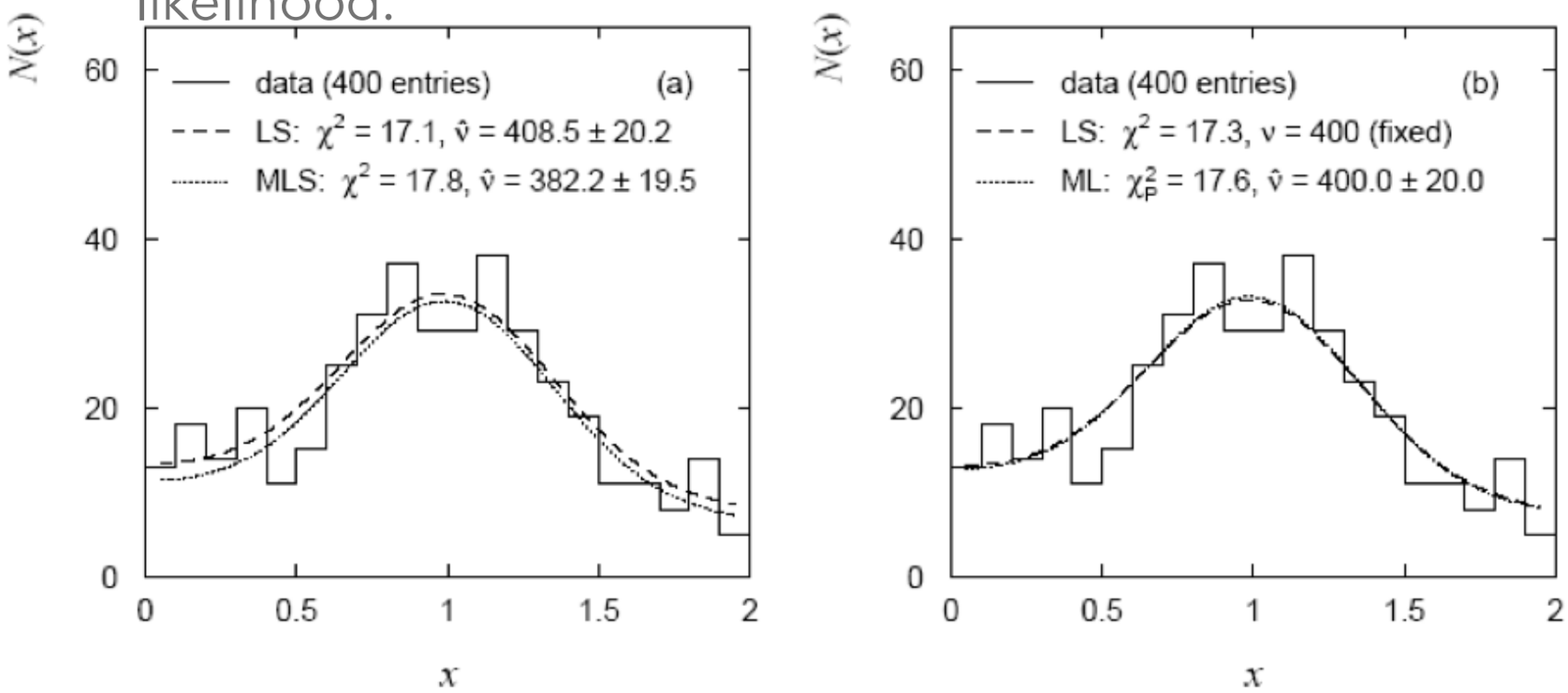
$$\lambda_i(\vec{\theta}, \nu) = \nu \int_{x_i^{min}}^{x_i^{max}} f(x; \vec{\theta}) dx = \nu p_i(\vec{\theta})$$

    - The estimator for n, $\hat{\nu}$, will be bad.

$$\hat{\nu}_{LS} = n + \frac{\chi^2_{min}}{2} \qquad \hat{\nu}_{MLS} = n - \chi^2_{min}$$

Method of Least Squares

- LS method with binned data
    - Normalization example: n=400 entries in N=20 bins.
    - The expected chi-square minimum is near N-m which means the relative error in the estimated normalization is large when N is large and n is small.
    - Ultimately get n directly from the data for LS method, or use a maximum likelihood.



Method of Least Squares

- LS method with binned data
    - Choices of binning is critical. Two common choices are:
        - equal width
        - equal probability
    - It is important not to choose the binning in order to make the chi-square minimum as small as possible! Doing so would cause the statistic to no longer follow the chi-square distribution.
    - It is necessary to have several entries (>5) in each bin so that the statistic approximates a standard normal distribution.

Method of Least Squares

- Combining measurements with LS method
    - The LS method may be used to obtain the weighted average of N measurements of the true value λ.
    - Given measurements, y, the variance, assumed to be known, is:

$$\sigma_i^2 = V[y_i]$$

    - For uncorrelated measurements:

$$\chi^2(\lambda) = \sum_{i=1}^{N} \frac{(y_i - \lambda)^2}{\sigma_i^2}$$

    - and, as usual, we solve for the first derivative equated to zero.

$$\hat{\lambda} = \frac{\sum_{i=1}^{N} y_i/\sigma_i^2}{\sum_{j=1}^{N} 1/\sigma_j^2} \qquad V[\hat{\lambda}] = \frac{1}{\sum_{i=1}^{N} 1/\sigma_i^2}$$

Method of Least Squares

- Combining measurements with LS method
    - If the covariance between measurements is $cov[y_i, y_j] = V_{ij}$, then minimize:

$$\chi^2(\lambda) = \sum_{i,j=1}^{N} (y_i - \lambda)(V^{-1})_{ij}(y_j - \lambda)$$

$$\hat{\lambda} = \sum_{i=1}^{N} w_i y_i \qquad\qquad w_i = \frac{\sum_{j=1}^{N}(V^{-1})_{ij}}{\sum_{k,l=1}^{N}(V^{-1})_{kl}}$$

$$V[\hat{\lambda}] = \sum_{i,j=1}^{N} w_i V_{ij} w_j$$

- The least square estimate has zero bias and minimum variance according to the Gauss-Markov theorem.

Method of Least Squares

- Using LS with biased data samples
  - It may happen that some data samples will not reflect the true distribution due to, for instance, unequal detection efficiency for each event. To deal with this it is best to modify the theoretical model to account for the detection efficiency. In doing so, no modification of the least squares minimization is necessary. If that is not possible you can either
    - Modify the events in a bin, n_i: If the detection efficiency for event j in bin i is:

$$\epsilon_{ij} \rightarrow n_i^{'} = \sum_{j=1}^{n_i} 1/\epsilon ij \qquad \text{and minimize} \qquad \chi^2 = \sum_{i=1}^{n_i} (n_i^{'} - f_i)^2 / f_i$$

    - Modify $f_i : \; f_i^{'} = f_i D_i \qquad D_i = n_i^{-1} \sum_{j=1}^{N} \epsilon_{ij}$

$$\textbf{and minimize} \quad \chi^2 = \sum_{i=1}^{N} (n_i - f_i^{'})^2 / f_i^{'}$$

  - Both work well when the variation of the weights is small, otherwise the uncertainty of the estimates are not well defined.