Today: Fundamentals of Monte Carlo

What is Monte Carlo?

- Named at Los Alamos in 1940's after the casino.

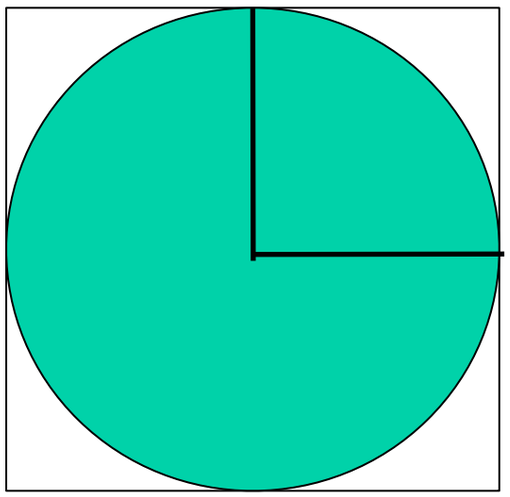
- Any method which uses *(pseudo)random numbers* as an essential part of the algorithm.

  Stochastic - not deterministic!

- A method for doing *highly dimensional integrals* by sampling the integrand.

- Often a Markov chain, called Metropolis MC.

Simple example: Buffon's needle
Monte Carlo determination of Pi



Consider a square inscribed by circle.
Consider one quadrant.

By geometry: $\dfrac{\text{area of 1/4 circle}}{\text{area of square}} = \dfrac{\pi r^2/4}{r^2} = \dfrac{\pi}{4}$

Simple MC like throwing darts at unit square target:

- Using RNG ~ (0,1), pick a pair of coordinates (x,y).
- Count # of points (darts) in shaded section versus total.

Pts in shaded circle/pts in square ~ $\pi/4$.

Code Snippet

```
hits = 0
for n in range(1,N):
    x = (random #)
    y = (random #)
    distance2 = (x² + y²)
    if (distance2 <= 1): hits += 1
pi= 4*hits/N
```

MC is advantageous for high dimensional integrals: the best general method

Consider an integral in the unit *D-dimensional* hypercube:

$$I = \int dx_1 \cdots dx_D \; f(x_1, \ldots, x_D)$$

By conventional deterministic methods:

- Lay out a grid with L points in each direction with **h=1/L**
- Number of points is **$N = L^D = h^{-D}$ ~ CPU time**.

How does error scale with CPU time or Dimensionality?

- Error in trapezoidal rule goes as $\varepsilon = f''(x)\, h^2$ since

$$\mathbf{f(x)=f(x_0) + f'(x_0)\, h + (1/2)f''(x_0)h^2 + \ldots}$$

- "direct" CPU time ~ $\varepsilon^{-D/2}$. (**$\varepsilon \sim h^2$ and CPU$\sim h^{-D}$**)
- But by sampling we find ε^{-2}. (**$\varepsilon \sim N^{-1/2}$ and CPU$\sim N$**)

To get another decimal place takes 100 times longer!

But MC is advantageous for D>4!

Improved Numerical Integration

Integration methods in D-dimensions:

- Trapeziodal Rule: $\varepsilon \sim f^{(2)}(x)\, h^2$
- Simpson's Rule: $\varepsilon \sim f^{(4)}(x)\, h^4$
- … **generally:** $\varepsilon \sim f^{(\alpha)}(x)\, h^{\alpha}$

And CPU time scales with sample points (grid size)

- **CPU time** $\sim h^{-D}$ (e.g., 1-D like 1/L and 2-D like $1/L^2$)
- **Time to do integral:** $T_{int} \sim \varepsilon^{-D/\alpha}$

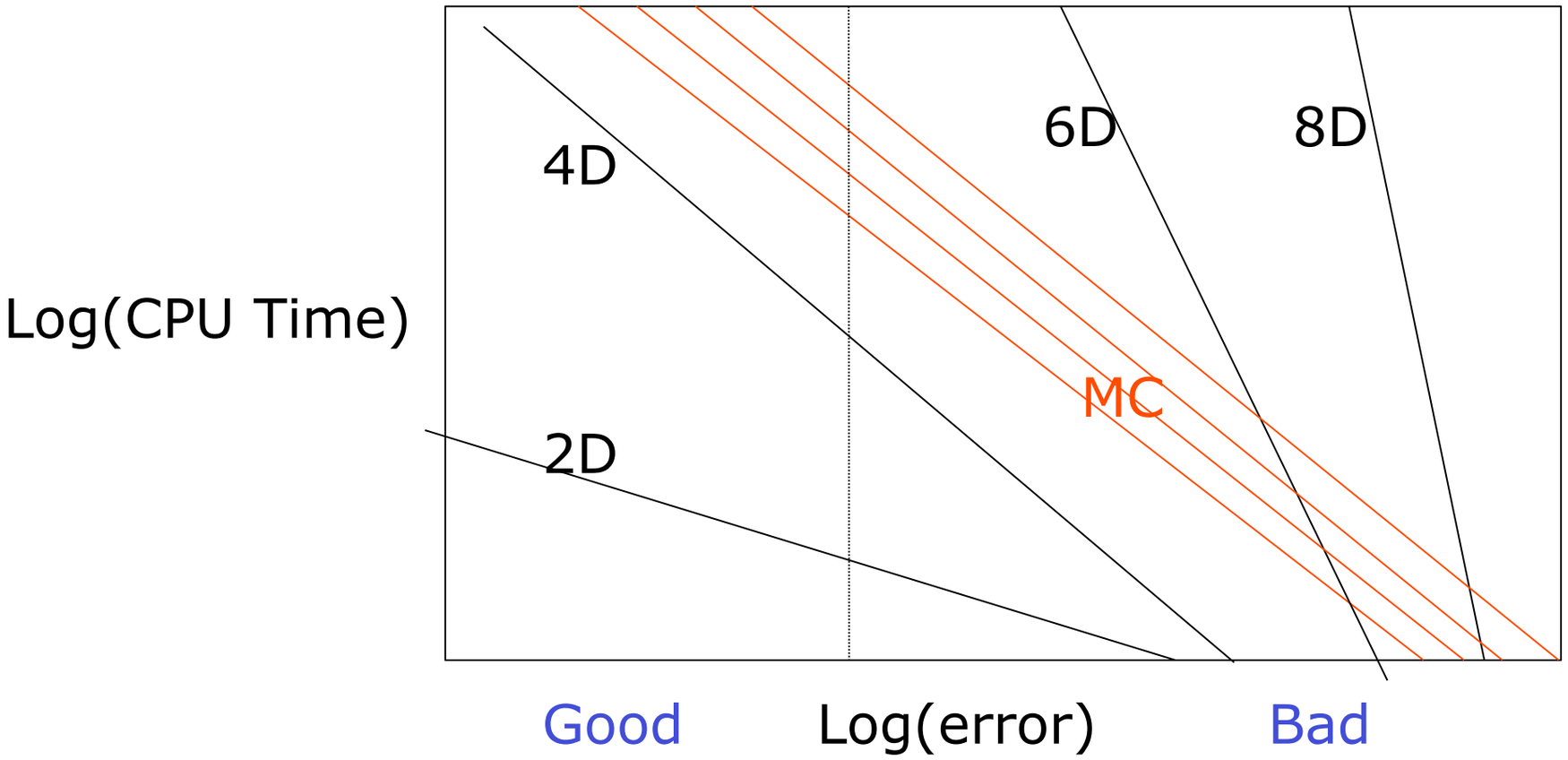
By conventional deterministic methods:

- Lay out a grid with L points in each direction with $h=1/L$
- Number of points is $N=h^{-D}=L^D \sim$ CPU time.

Stochastic Integration (Monte Carlo)

- Monte Carlo: $\varepsilon \sim M^{-1/2}$ (sqrt of sampled points)
- CPU time: $T \sim M \sim \varepsilon^{-2}$.

> In the limit of small ε, MC wins if $D > 2\alpha$!



Other reasons to do Monte Carlo:

- Conceptually and practically simple.
- Comes with built in error bars.

Many methods of integration have been tried, and will be tried in this world of sin and woe. No one pretends that Monte Carlo is perfect or all-wise. Indeed, it has been said that Monte Carlo is the worst method except all those other methods that have been tried from time to time. Churchill 1947

Probability Distributions

- P(x)dx= probability of observing a value in (x,x+dx) is a probability distribution function (p.d.f.)

$$\int dx\, P(x) = 1 \qquad P(x) \geq 0$$

- x can be either a continuous or discrete variable.
- Cumulative distribution:

  Probability of x<y.

  Useful for sampling

$$C(y) = \int_{-\infty}^{y} dx\, P(x) \qquad 0 \leq C(y) \leq 0$$

- Average or expectation

$$\langle g(x) \rangle = \bar{g} = \int_{-\infty}^{\infty} dx\, P(x) g(x)$$

- Moments:
  - Zeroth moment $I_0=1$
  - Mean $\langle x \rangle = I_1$
  - Variance $\langle (x-\langle x \rangle)^2 \rangle = I_2-(I_1)^2$

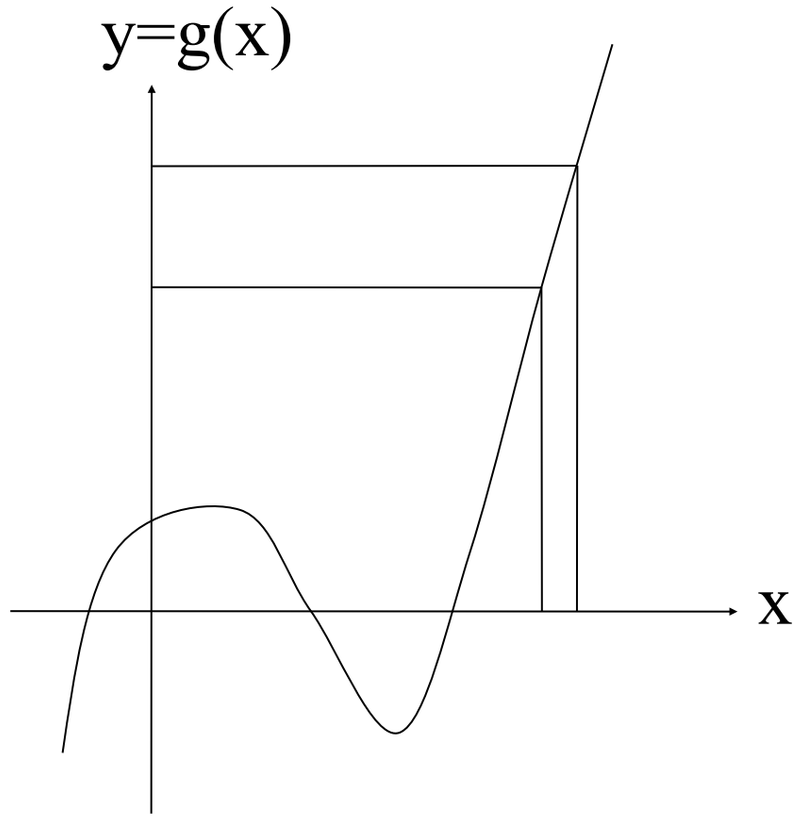
$$I_n = \langle x^n \rangle = \int_{-\infty}^{\infty} dx\, P(x) x^n$$

Mappings of random variables

Let $\mathbf{p_x(x)dx}$ be a probability distribution

Let **y=g(x)** be a new variable

-- e.g., y=g(x) = –ln(x)
with $0 < x \leq 1$, so $y \geq 0$



What is the pdf of y?

With $p_y(y)dy=p_x(x)dx$

$$p_y(y) = p_x(x)\left|\frac{dx}{dy}\right| = p_x(x)\left|\frac{dx}{dg(x)}\right| = p_x(x)\left|\frac{dg(x)}{dx}\right|^{-1}$$

What happens when g is not monotonic?

Example: y=g(x) = –ln(x)

$$p_y(y)dy = p_x(x)\left|\frac{dg(x)}{dx}\right|^{-1} dy = e^{-y}dy$$

*Distributed exponentially, like in Poisson event, e.g. y/λ has PDF of $\lambda e^{-\lambda y}$

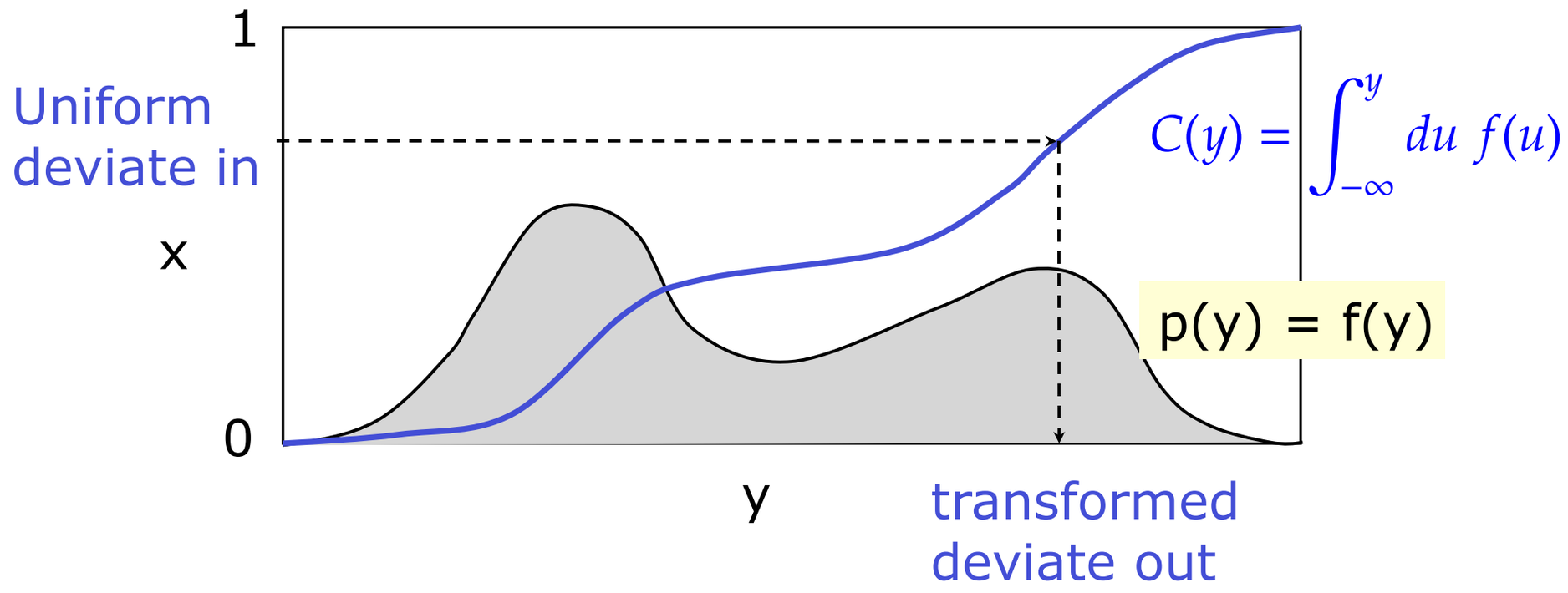
What is the Mapping Doing?

Generate random deviate

$$p(x)dx = \begin{cases} dx & 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases} \qquad \int_{-\infty}^{\infty} p(x)\,dx = 1$$

PDF is normalized

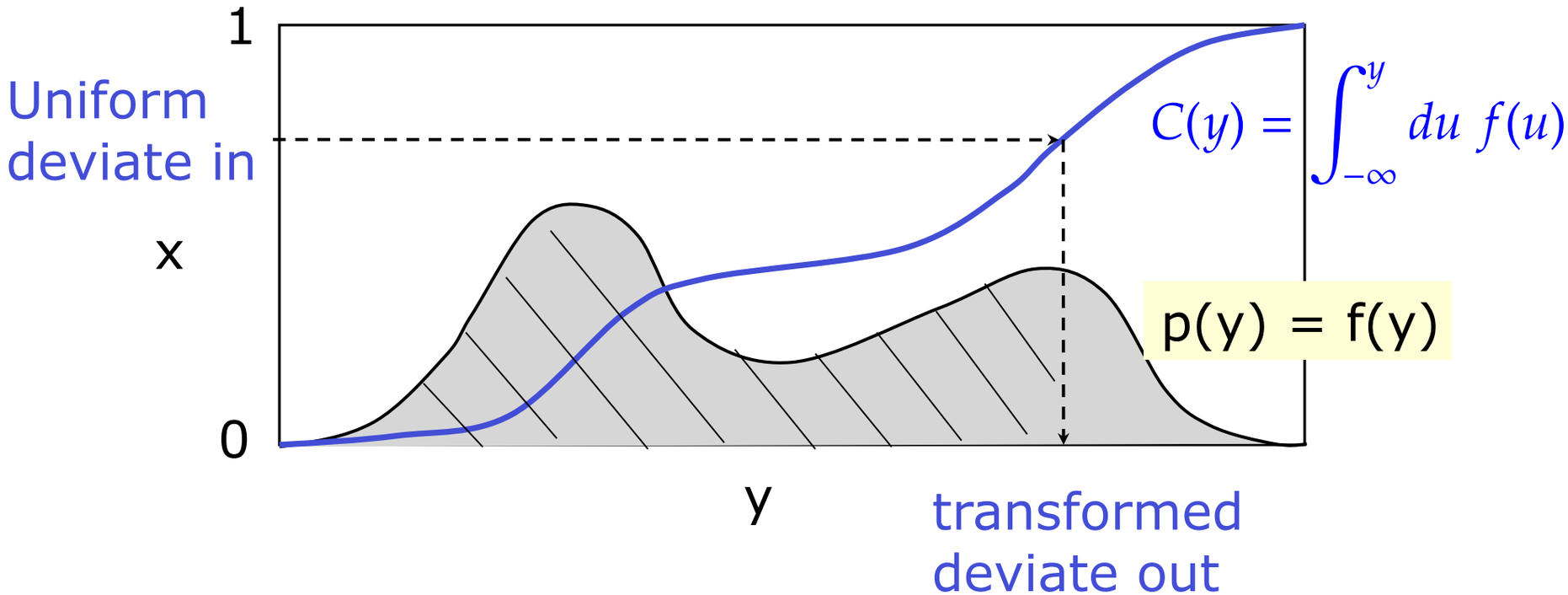
Let p(y)=f(y), then y(x) = $C^{-1}(x)$ a functional inverse of C(y).
Thus, dx/dy = f(y) or x=C(y)



- Allows a random deviate y from a known probability distribution p(y).
- The indefinite integral of p(y) must be known and invertible.
- A uniform deviate x is chosen from (0,1) such that its corresponding y on the definite-integral curve is desired deviate, i.e. x = c(y).

Interpreting the Mapping

Let $p(y)=f(y)$, then $y(x) = C^{-1}(x)$, $dx/dy = f(y)$ and $x=C(y)$



Since C(y) is area under p(y)=f(y), $y(x) = C^{-1}(x)$ prescribes that

- Choose x=(0,1], then find value y that has that fraction x of area to left of y, or x=C(y)
- Return that value of y.

Example: Drawing from Poisson Distribution

$y= -\ln(x)/\lambda$, $p(y)= \lambda e^{-\lambda y}$ and $x=F(y) = 1 - e^{-\lambda y}$

Note: $x = 1 - e^{-\lambda y}$ or $x' = 1- x = e^{-\lambda y}$ (with x'=(0,1]).

So, indeed, $y(x) = -\ln(x')/\lambda$

Example: Drawing from Normal Gaussian

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

Box-Muller method: get random deviates with normal distr. $p(y)dy = \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy$

Consider: uniform deviates (0,1), x_1 and x_2, and two quantities y_1 and y_2.

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

Or, equivalently,

$$x_1 = \exp\left(-\frac{1}{2}\left[y_1^2 + y_2^2\right]\right) \qquad x_2 = \frac{1}{2\pi}\tan^{-1}\left(\frac{y_2}{y_1}\right)$$

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

So, each y is independently normal distributed.

Better: Pick $R^2 = v_1^2 + v_2^2$ so $x_1 = \sqrt{R^2}$ and $\tan^{-1}(v_2/v_1) = 2\pi x_2$

Advantage: no sine and cosine by using $v_1/\sqrt{R^2}$ and $v_2/\sqrt{R^2}$and
And get two RNG per calculation (1 for now, 2 for next time)