Sampling Distributions

Allen & Tildesley pgs. 345-351
and "Numerical Recipes" on random numbers

Today we generate ***non-uniform probability distributions***.

These are used in *importance sampling* to reduce the variance of a Monte Carlo evaluation or to simulate various physical processes.

We start by assuming that there is software to generate udrn's (uniform RN) in the range (0,1).

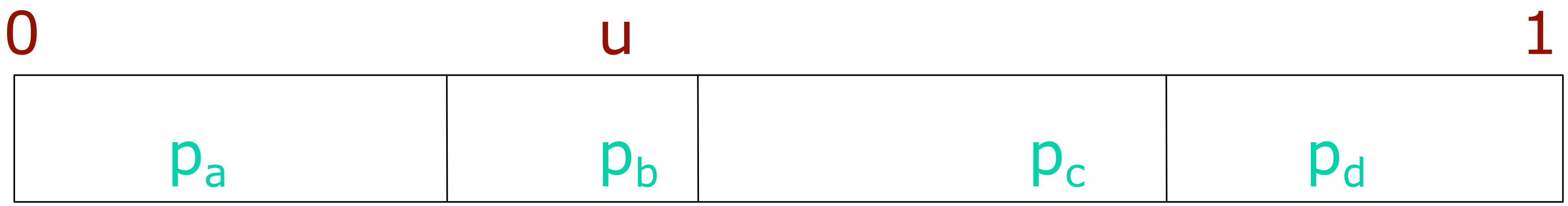
How do we sample an arbitrary p(x)dx ?

There are lots of tricks.

Discrete Distributions

Any discrete distribution p_k can be sampled by constructing the cumulative distribution

$$c_k = \sum_{i=1}^{k} p_i$$



- **Sample 0<u<1**.
- Find which region it is in, i.e. **find k**: $\mathbf{c_{k-1} < u < c_k}$
- **Return label "k"**.

- The search can be done by ***bisection*** in $\log_2(N)$ steps.
- For simple distributions, it might be even easier.
- There is a faster more complicated method (see notes).

Make a Table $\mathbf{c_k = c_{k-1} + p_k}$ before simulation.

Continuous Distributions

Generalize this to a continuous function.

This is the mapping method.

- Construct the cumulative probability distribution:

$$C(y) = \int_{-\infty}^{y} dx\, p(x)$$

- Sample u in (0,1)
- Find $x=c^{-1}(u)$ *An inverse mapping from urnd to x.*

x is sampled from p(x) dx, which is non-uniform.

Can always perform this with a table lookup.

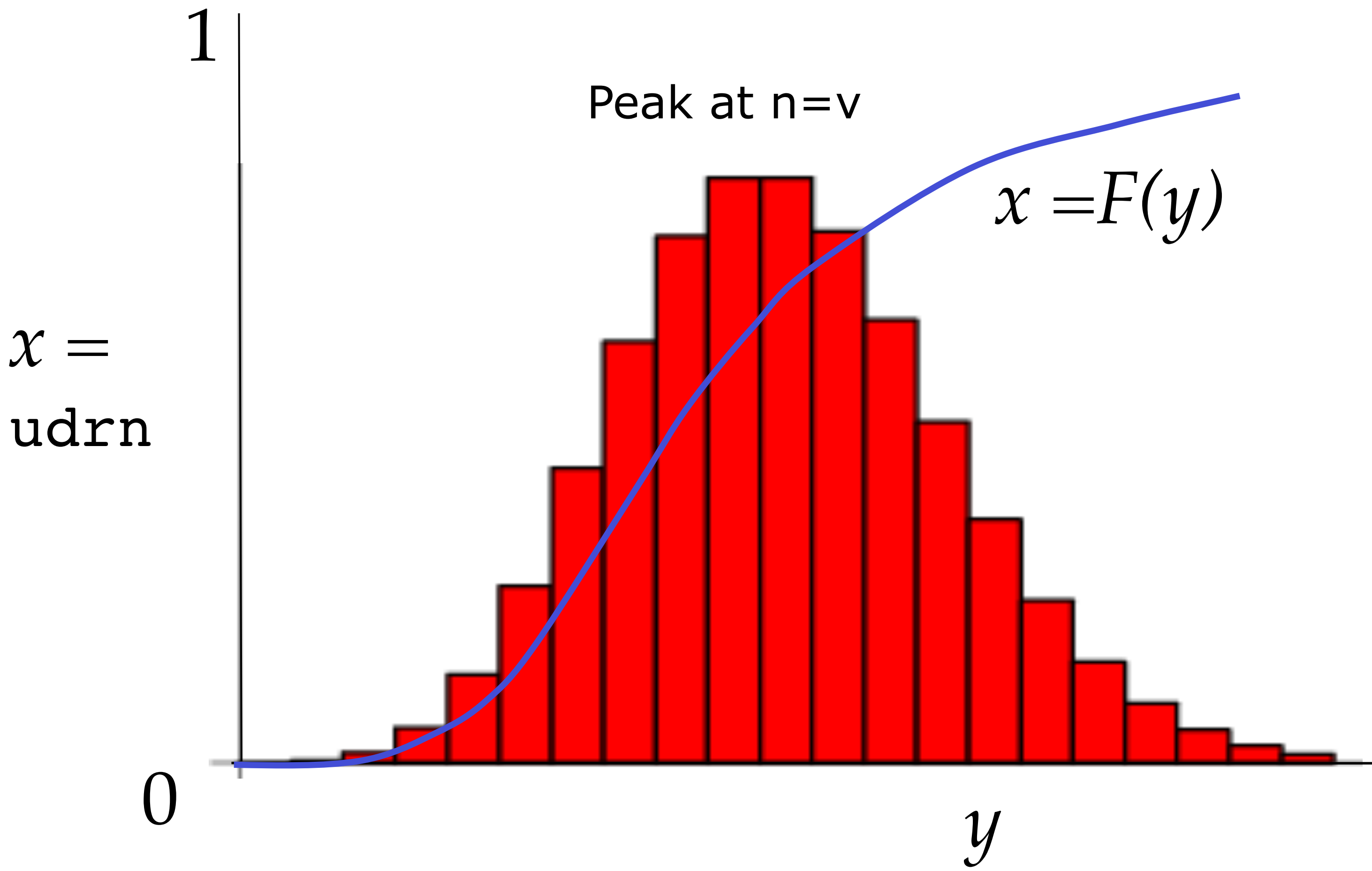
Some analytic examples:

- $P(x)=a\ e^{-ax}$ $0 < x$ then $x = -\ln(u)/a$
- $P(x)=(a+1)x^{a}$ $0 < x < 1$ then $x = u^{1/(a+1)}$

Problem if inverse mapping is difficult.

Example: Drawing from Poisson Distribution

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



For n successes in N trials

$$P_B(n|N) = \frac{N!}{n!(N-n)!}p^n(1-p)^{N-n}$$

As fct of expected successes v=Np

$$P_{v/N}(n|N) = \frac{N!}{n!(N-n)!}\left(\frac{v}{N}\right)^n\left(1-\frac{v}{N}\right)^{N-n}$$

$$\lim_{N\to\infty} : \frac{v^n e^{-v}}{n!}$$

As F(y) is area under p(y), y(x) = F^{-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=F(y)
- Return that value of y.

Rejection technique

- Sample x from $q(x)dx$
- Accept x with probability $c(x)$, otherwise repeat

What is distribution of accepted x's?

$p(x)dx = c(x)q(x)dx/\text{normalization}$

Hence choose $c(x) = a\, p(x)/q(x)$

Where **a** is set so that $c(x) \leq 1$ so $a \leq \min[q(x)/p(x)]$

1/a is the *acceptance probability*, the *efficiency*.

– Do not *under sample*, or else efficiency will be low.

– If inefficient, you will use lots of prns/step.

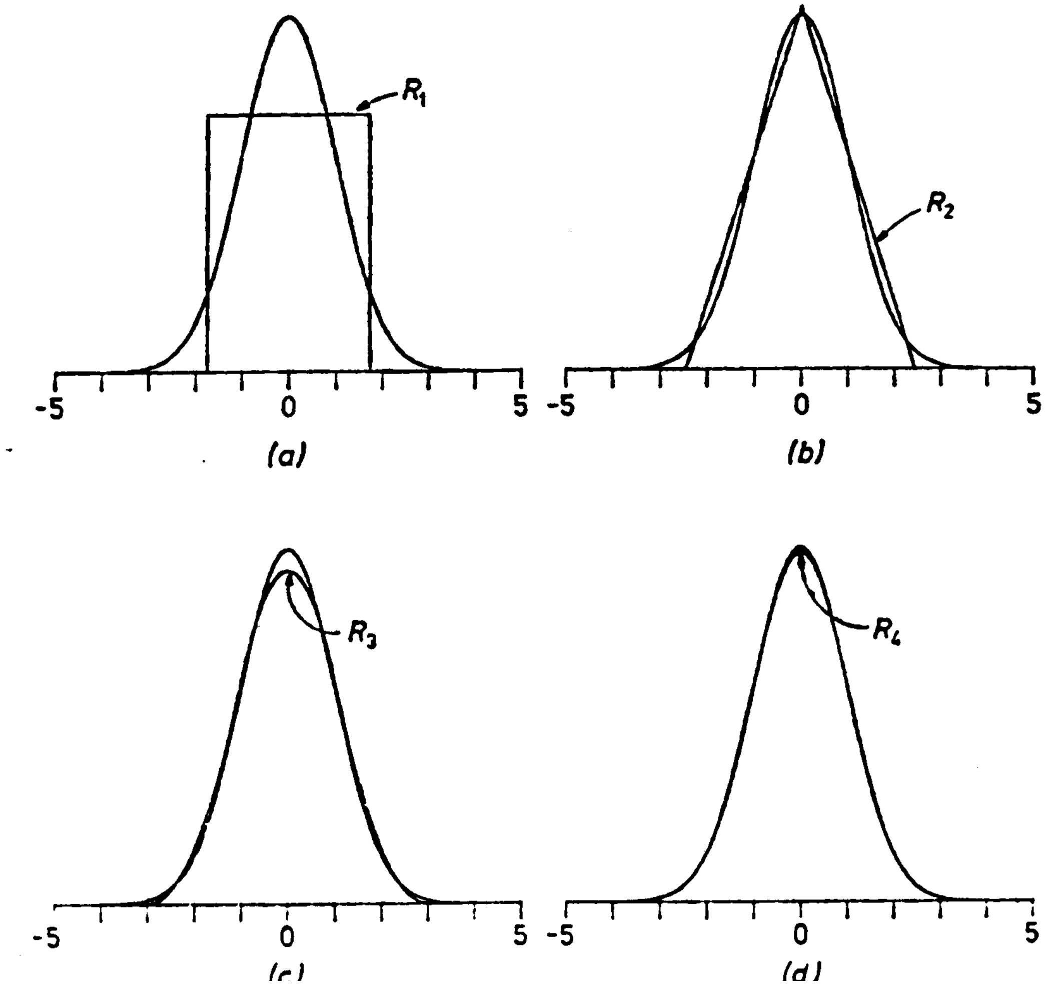
Composition method

Combine several random numbers

- **Add several udrn's.**

  remember characteristic function

  limit for large k is a Gaussian



Example: add two integers in (1,6)

Example: add 12 udrn, i.e

$$\zeta = \sum_{i=1}^{12} u_i - 6, \quad u_i \in [0,1]$$

Gaussian distributed on (-6,6).

- Take maximum of 'k' udrns.

  x= max(u_1, .., u_k) Prove that P(x)= k x^{k-1}

Normal distribution

- Inverse mapping is a little slow, also of infinite range.
- Trick: Generate 2 prns at a time: **r**=(y_1,y_2)
  - Generate x_1 and x_2
  - Calculate $y_1 = \sqrt{-\ln x_1}\cos 2\pi x_2$ $\quad y_2 = \sqrt{-\ln x_1}\sin 2\pi x_2$

Samples:

$$p(y_1, y_2)\, dy_1 dy_2 = (2\pi)^{-1}\exp(-r^2/2) = p(r)r\, dr d\theta$$

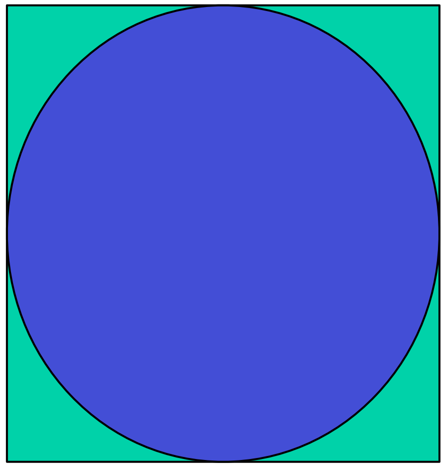
$$p(v)\, dv = \frac{1}{2}\exp(-v/2)\, dv, \quad \text{with } v = r^2$$

We can sample the angle using a *rejection technique*:

Sample (x,y) in square

Accept if $x^2+y^2 <1$

Normalize to get the correct r.



Example: Box-Mueller Code to sample a Normal Distribution

Normal distribution $\langle x \rangle = x_0$ and $\langle (x-x_0)^2 \rangle = \sigma^2$

```
while r2 > 0.25:
    x,y = sprng()-0.5,sprng()-0.5
    r2 = x*x+y*y
radius = sigma*sqrt(-2*ln(sprng())/r2)
xnormal = x0 + x*radius
ynormal = y0 + y*radius
```

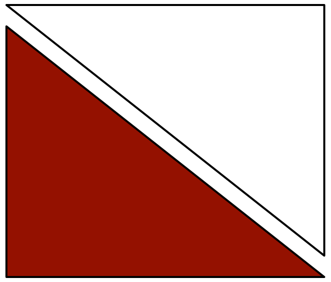
- No trig functions!
- Rejection mixes up regularity of random numbers
- Efficiency of angle generation is $4/\pi$.
- Will get 2 ndrn’s each time.

Multivariate normal distributions

How to sample a correlated Gaussian? (say with D components)

- Assume we want $<x_i x_j> = T_{ij}$
- Take the square root of T : $SS^{T} = T$

Make “Choleski” decomposition (see Numerical Recipes or notes).
We can assume S is a triangular matrix
all entries above diagonal are zero.



- Generate D normally distributed numbers y.
- Transform to correlated random distribution

$$x_\alpha = \sum_i S_{\alpha i} y_i, \quad \langle y_i y_j \rangle = \delta_{ij}$$

$$\langle x_\alpha x_\beta \rangle = \sum_{ij} \left\langle S_{\alpha i} y_i S_{\beta j} y_j \right\rangle = \sum_{ij} S_{\alpha i} S_{\beta j} \delta_{ij} = SS^{\mathrm{T}} = T$$

Reduction of Variance

As we discussed earlier, the statistical error goes as:
error = sqrt(variance/computer time).

DEFINE:

$$\text{Efficiency} = \zeta = 1/vT$$

$v = \text{error}^2$ of mean and $T = $ total CPU time

How can you make simulation more efficient?

- Write a faster code,
- Get a faster computer
- Work on *reducing the variance*.
- Or all three

We will talk about the third option:
Importance sampling and *correlated sampling*

Importance Sampling

Given the integral:

$$I = \int dx\ f(x) = \langle f(x) \rangle$$

How should we sample *x* to maximize the efficiency?

Transform the integral:

$$I = \int dx\ p(x) \left(\frac{f(x)}{p(x)} \right) = \left\langle \frac{f(x)}{p(x)} \right\rangle_p$$

Estimator

variance is:

$$v \approx \sigma^2 = \left\langle \left(\frac{f(x)}{p(x)} - I \right)^2 \right\rangle_p = \int dx\ p(x) \left(\frac{f(x)}{p(x)} \right)^2 - I^2$$

Optimal sampling:

$$\frac{\delta v}{\delta p(x)} = 0 \quad \text{with} \quad \int dx\ p(x) = 1$$

Mean value of estimator *I is independent of p(x), but variance v is not*!
Assume CPU-time/sample is independent of p(x), and vary p(x) to minimize v.

Finding Optimal p*(x) for Sampling

Trick to parameterize as a positive definite PDF: $p(x) = \dfrac{[q(x)]^2}{\int dx\, [q(x)]^2}$

Solution: $p^*(x) = \dfrac{|f(x)|}{\int dx\, |f(x)|}$ Estimator: $\dfrac{f(x)}{p^*(x)} = \dfrac{\mathrm{sign}(f(x))}{\int dx\, |f(x)|}$

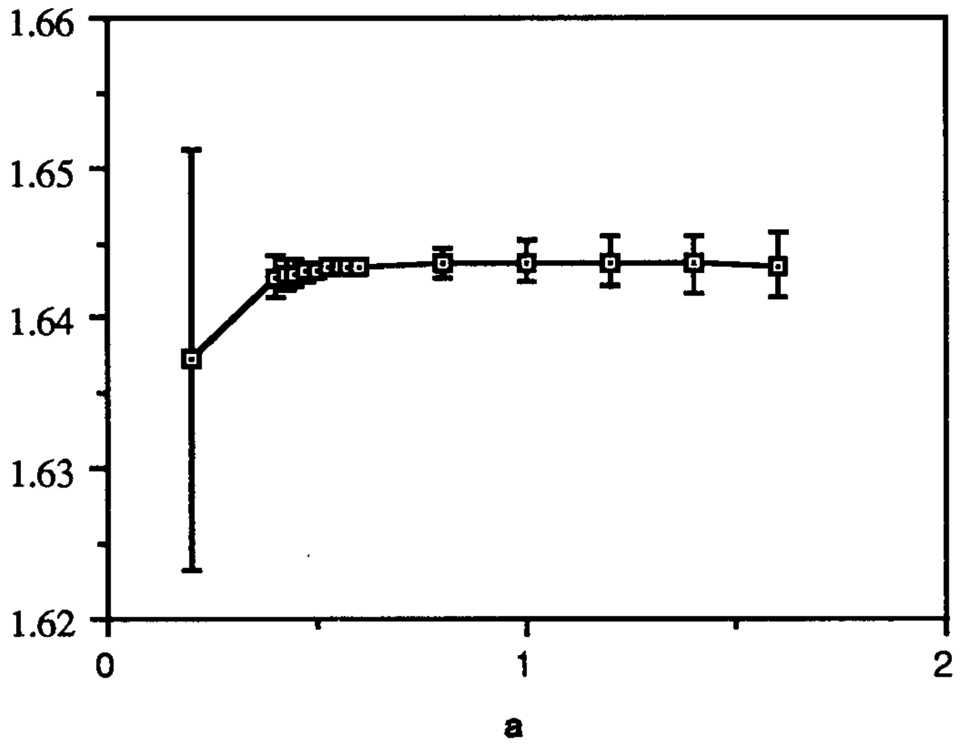
1. If f(x) is entirely positive or negative, estimator is constant. "zero variance principle."
2. We can't generally sample p*(x), because, if we could, then we would have solved problem analytically! But the *form* of *p*(x) is a guide to lowering the variance*.
3. Importance sampling is a general technique: *it works in any dimension*.

Example of importance sampling

Suppose f(x) was given by

$$f(x) = \frac{e^{-x^2/2}}{1+x^2}$$

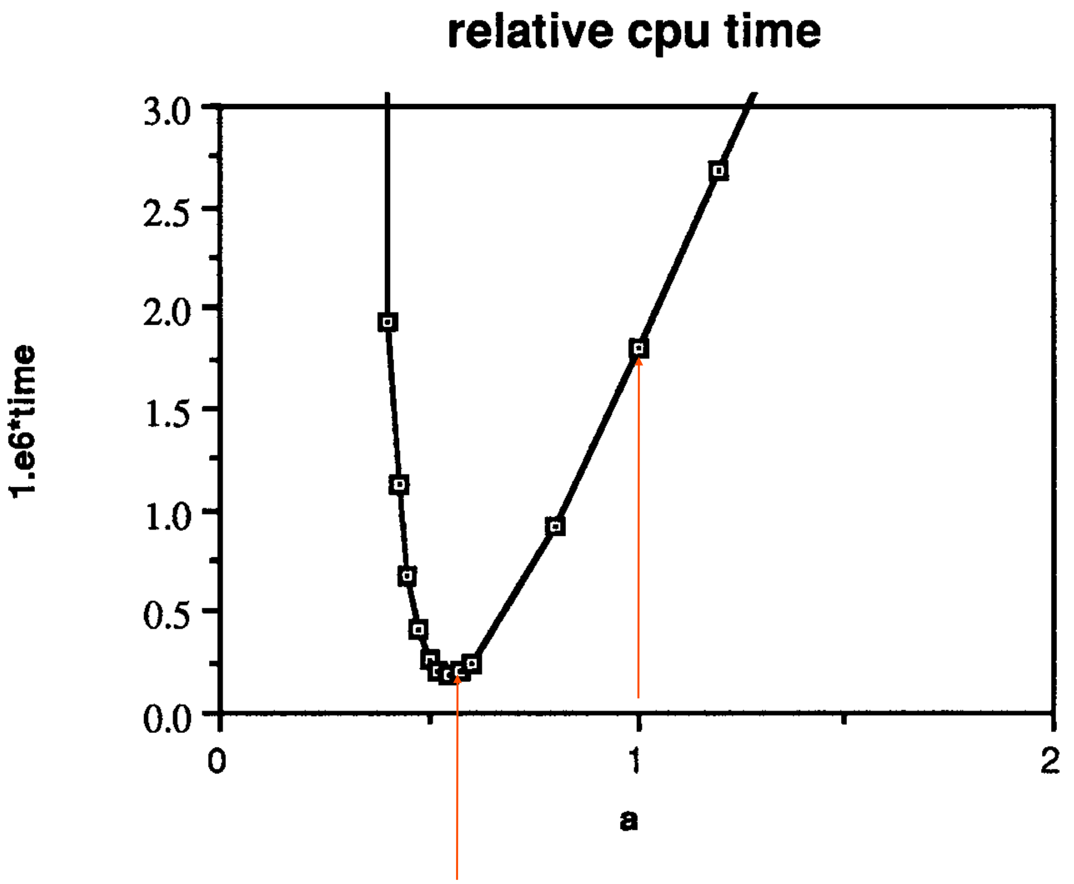
Value is independent of a.



Optimize a Gaussian

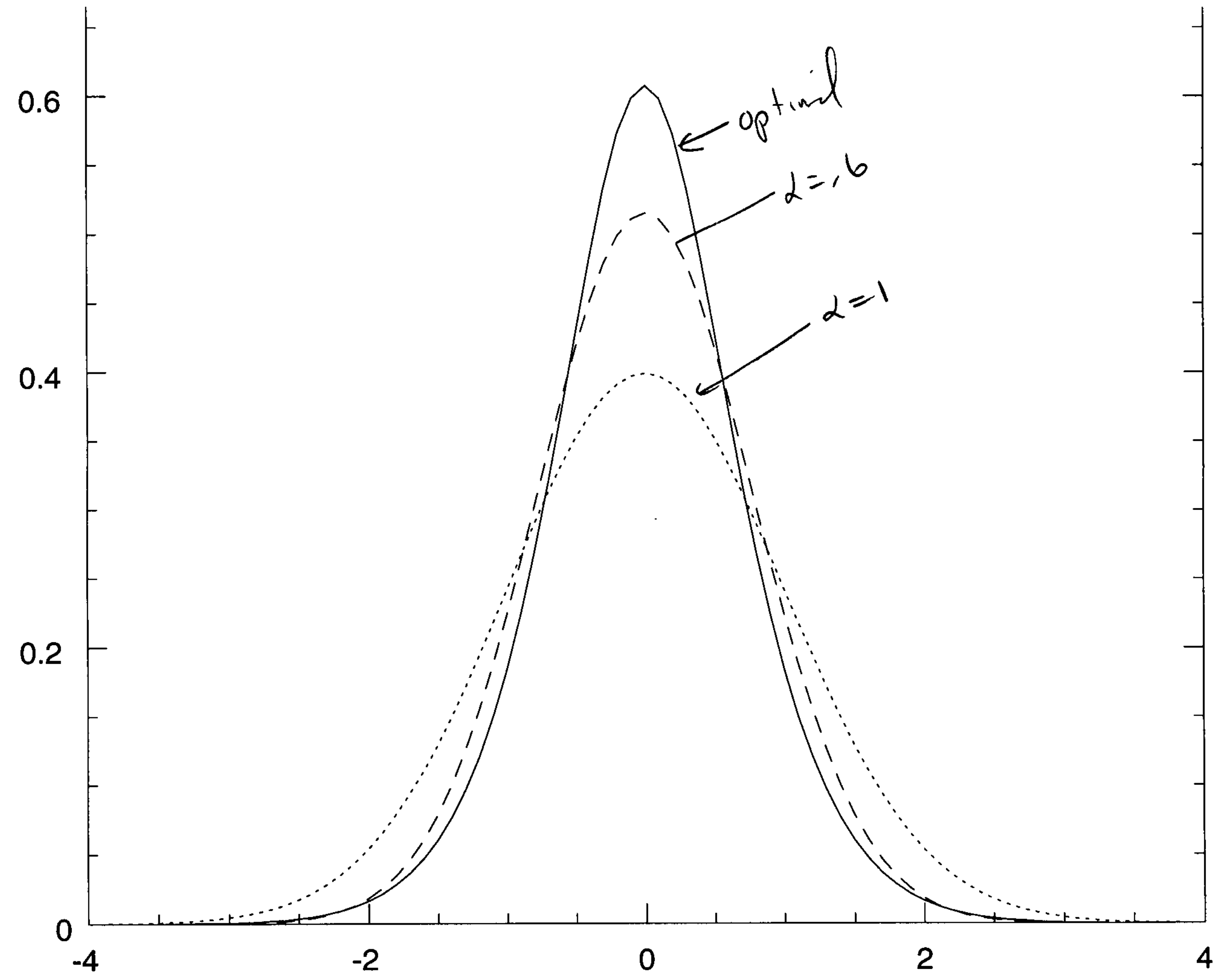
$$p(x) = \frac{e^{-x^2/2a}}{\sqrt{2\pi a}}$$

CPU time is not

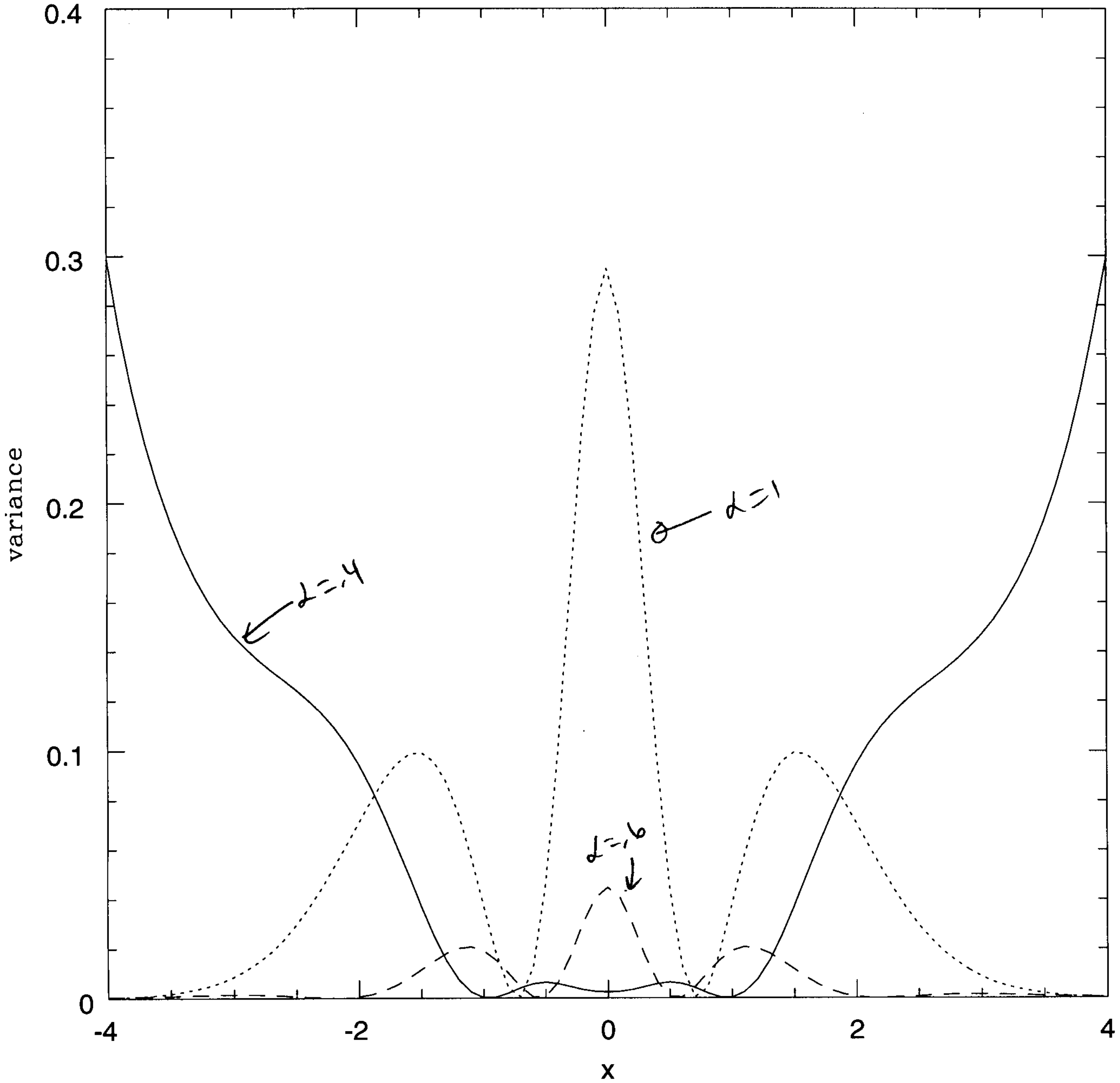


$$v \approx \sigma^2 = \left\langle \left(\frac{f(x)}{p(x)} - I \right)^2 \right\rangle_p = \int dx \; \frac{\sqrt{2\pi a}\; e^{-x^2(1-1/2a)}}{(1+x^2)^2} \; - I^2$$

Importance sampling functions

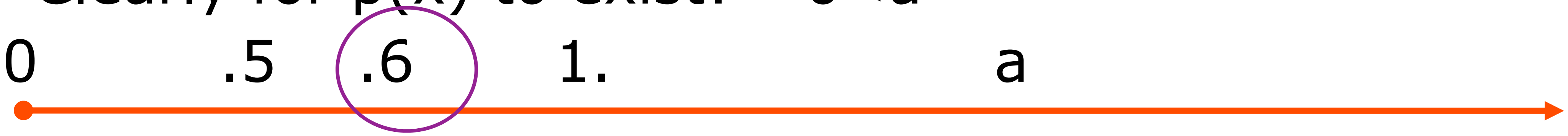


Variance integrand



What are allowed values of a?

- Clearly for p(x) to exist: 0<a



- For finite estimator 1<a

- For finite variance .5<a

- “Obvious” value a=1

- **Optimal value** a=0.6.

$$\sigma^2 = \int dx \, \frac{\sqrt{2\pi a}}{(1+x^2)^2} e^{-x^2(1-1/2a)} - I^2$$

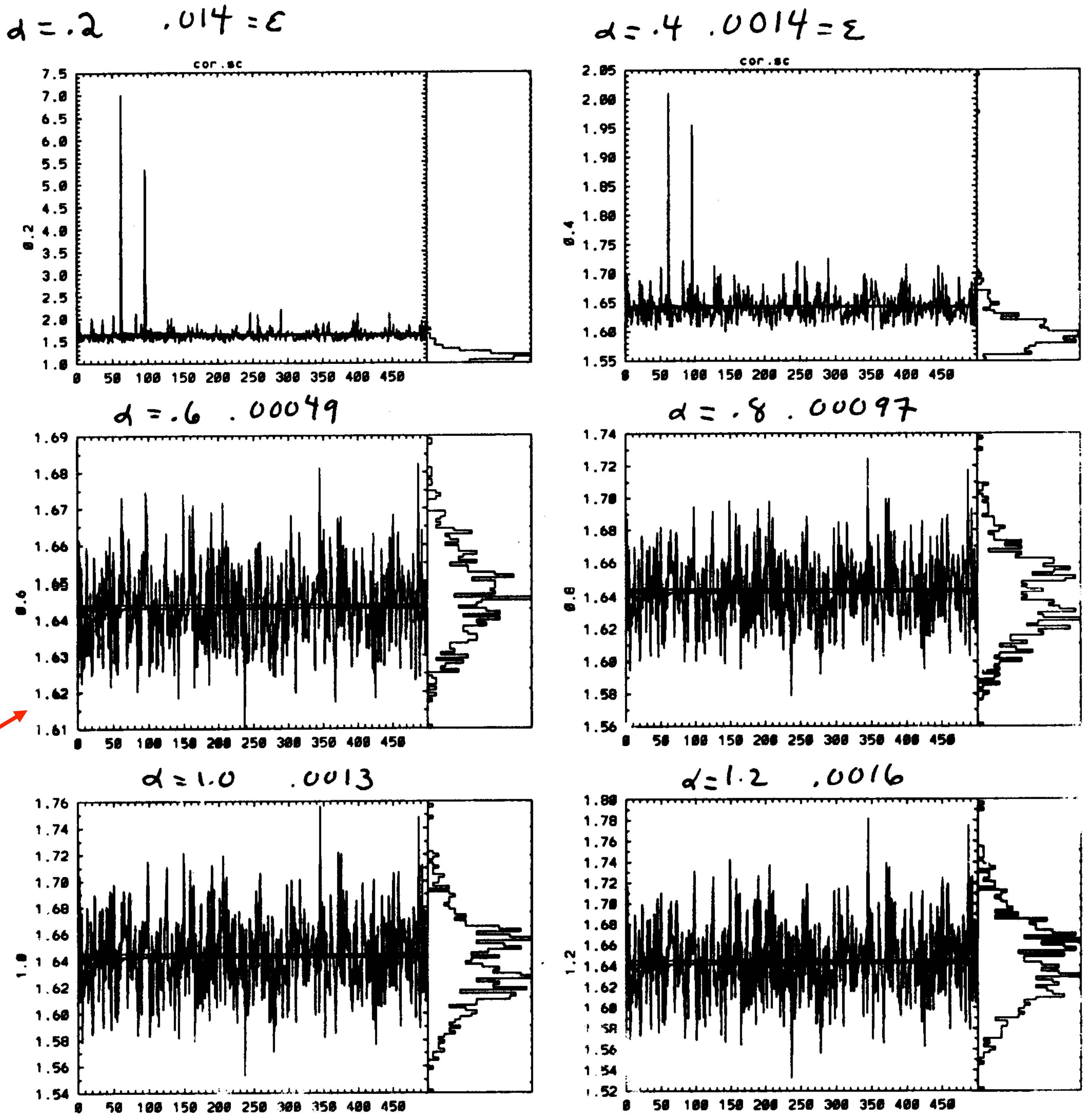
What does infinite variance look like?

Spikes

Long tails on the distributions

Near optimal

Why (visually)?



General Approach to Importance Sampling

- Basic idea of importance sampling is to sample more in regions where function is large.
- Find a convenient approximation to |f(x)|.

- ***Do not under-sample*** -- could cause infinite variance.
- ***Over-sampling*** -- loss in efficiency but not infinite variance.
- *Always derive conditions for finite variance analytically.*
- To debug: *test that estimated value is independent of important sampling*.

• *Sign problem*: zero variance is not possible for oscillatory integral.

"Monte Carlo can add but not subtract."

Correlated Sampling

Suppose we want to compute a function : $G(F_1,F_2)$

where F's are integrals: $F_k = \int dx\, f_k(x)$

Suppose we use same p(x) and same random numbers to do both integrals.

What is optimal p(x)?

$$p^*(x) \propto \left| f_1(x)\frac{\partial G}{\partial F_1} + f_2(x)\frac{\partial G}{\partial F_2} \right|$$

It is a weighted average of the distributions for F_1 and F_2.

Sampling Boltzmann distribution

- Suppose we want to calculate a whole set of averages:

$$\langle A_i \rangle = \frac{\int dr\, A_i(r) e^{-\beta V(r)}}{\int dr\, e^{-\beta V(r)}}$$

- Optimal sampling is: $p_i^*(r) \propto \left| e^{-\beta V(r)} (A_i(r) - \langle A_i \rangle) \right|$

independent of A

dependent on A

- We need to sample the first factor

  because we want lots of properties
- **Avoid undersampling**.
- The Boltzmann distribution is *very highly peaked*.

Independent Sampling for exp(-V/kT)?

- Try hit or miss MC to get Z= exp(-V/kT).
- Sample R uniformly in (0,L): P(R) = Ω^{-N}=1

What is the variance of the free energy and how does it depend on the number of particles?

$$\mathrm{var}(Z) = \int dr \left(e^{-\beta V(r)} - Z(\beta)\right)^2 = Z(2\beta) - Z(\beta)^2$$

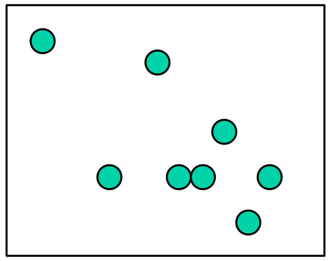
$$\mathrm{var}(\beta F) = \frac{\mathrm{var}(Z)}{Z^2} = \frac{Z(2\beta)}{Z(\beta)^2} - 1 = e^{2\beta(F(\beta)-F(2\beta))} - 1$$

$$\mathrm{error}(\beta F) = \exp\left[\beta(F(\beta) - F(2\beta))\right]$$

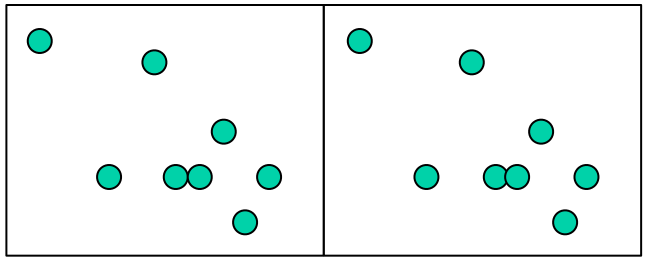
- Blows up exponentially fast at large N since F is extensive!
- The number of sample points must grow exponentially in N, just like a grid based method.

Intuitive explanation

- Throw N points in a box, area A.
- Say *probability of no overlap is q*.



- Throw 2N points in a box, area 2A.
- Probability of no overlap is q^2.



- Throw mN points in a box, area mA
- Probability of no overlap is q^m.

Probability of getting a success is p=exp(m ln(q)). Success defined as a reasonable sample of a configuration.

This is a general argument. We need to sample only near the peak of the distribution: to do so use random walks.