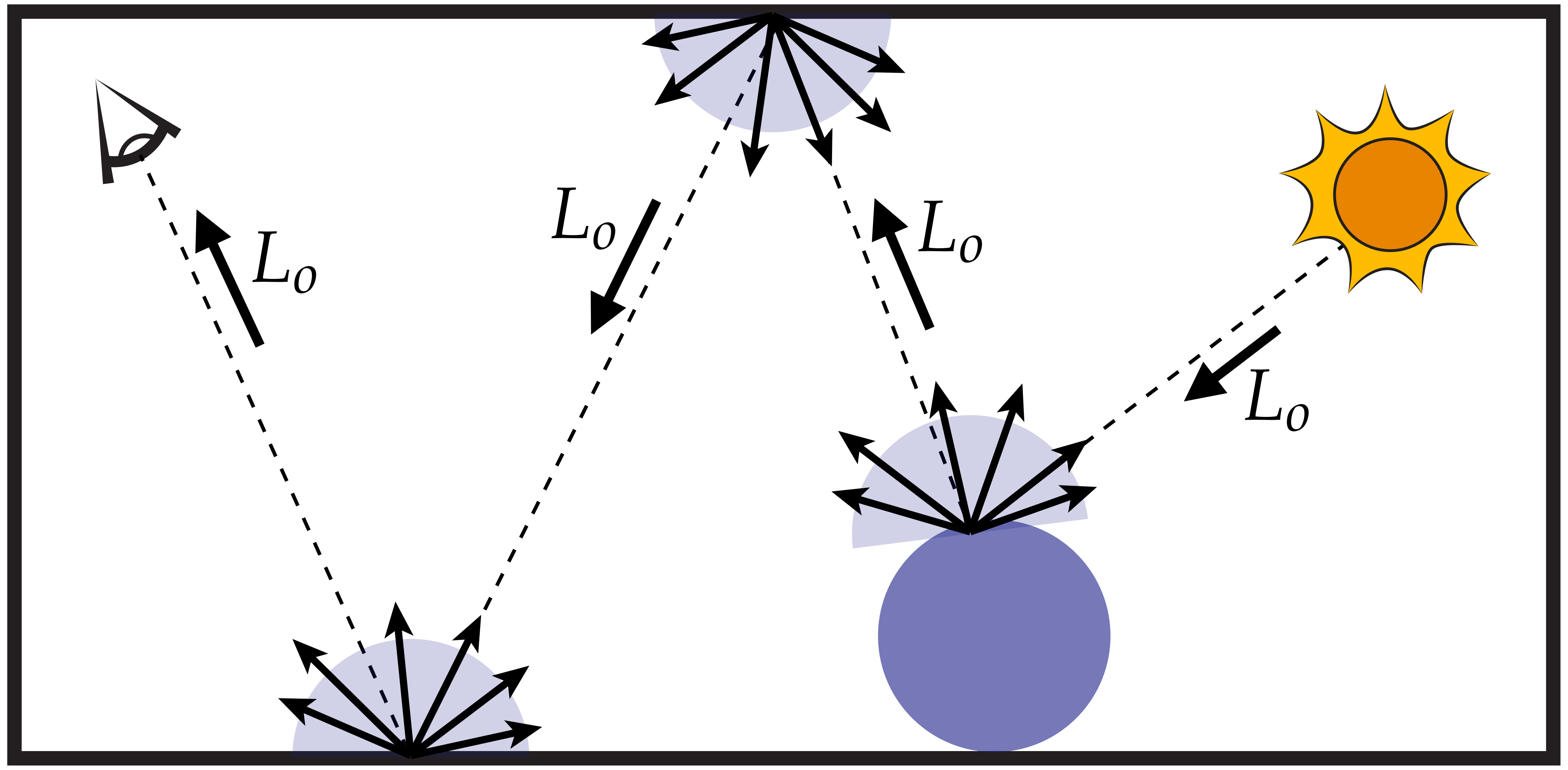
Numerical Integration

Computer Graphics
CMU 15-462/15-662

Motivation: The Rendering Equation

- **Recall the rendering equation, which models light "bouncing around the scene":**



$$L_o(\mathbf{p},\omega_o) = L_e(\mathbf{p},\omega_o) + \int_{\mathcal{H}^2} f_r(\mathbf{p},\omega_i \to \omega_o) L_i(\mathbf{p},\omega_i)\cos\theta\, d\omega_i$$

How can we *possibly* evaluate this integral?

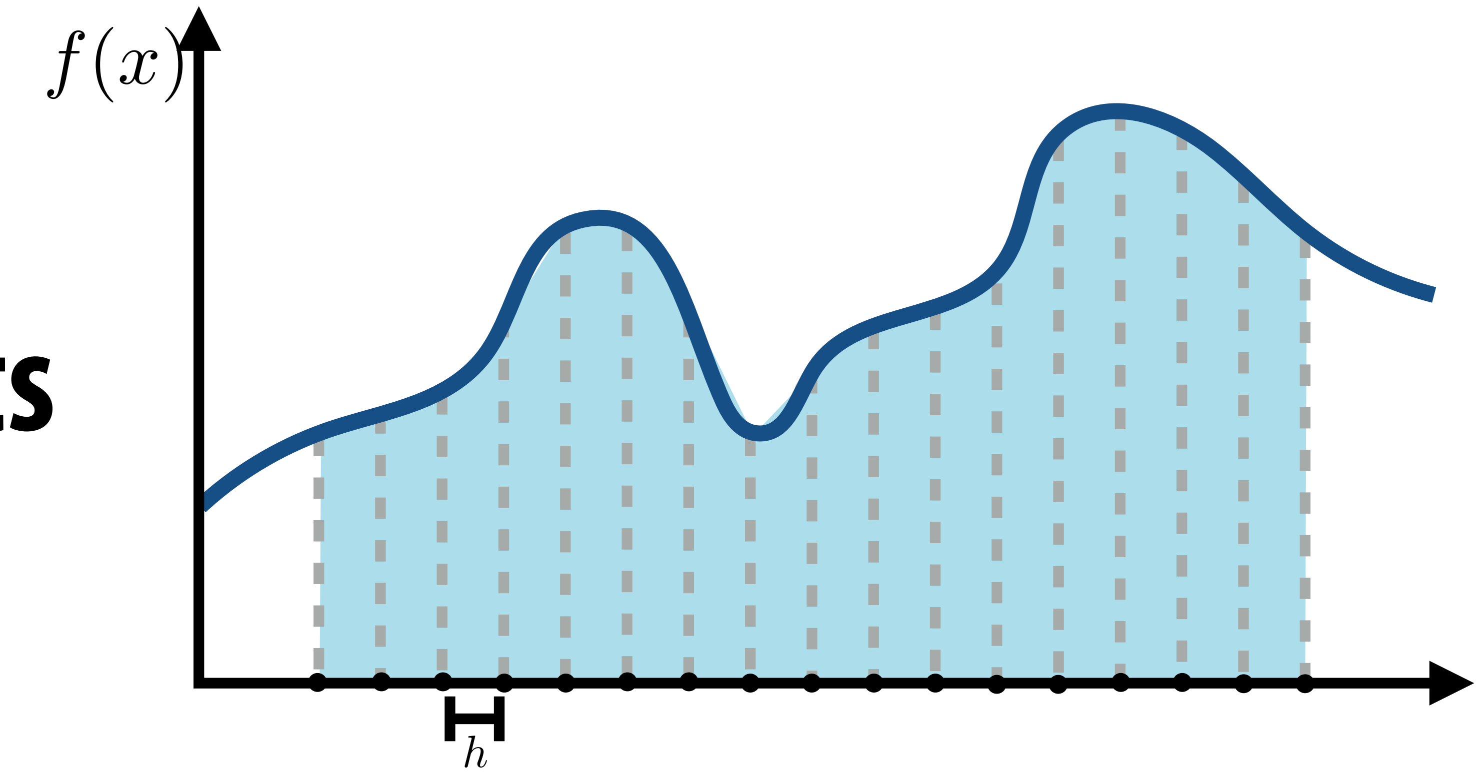
Numerical Integration—Overview

- In graphics, many quantities we're interested in are naturally expressed as integrals (total brightness, total area, ...)

- For very, *very* simple integrals, we can compute the solution analytically

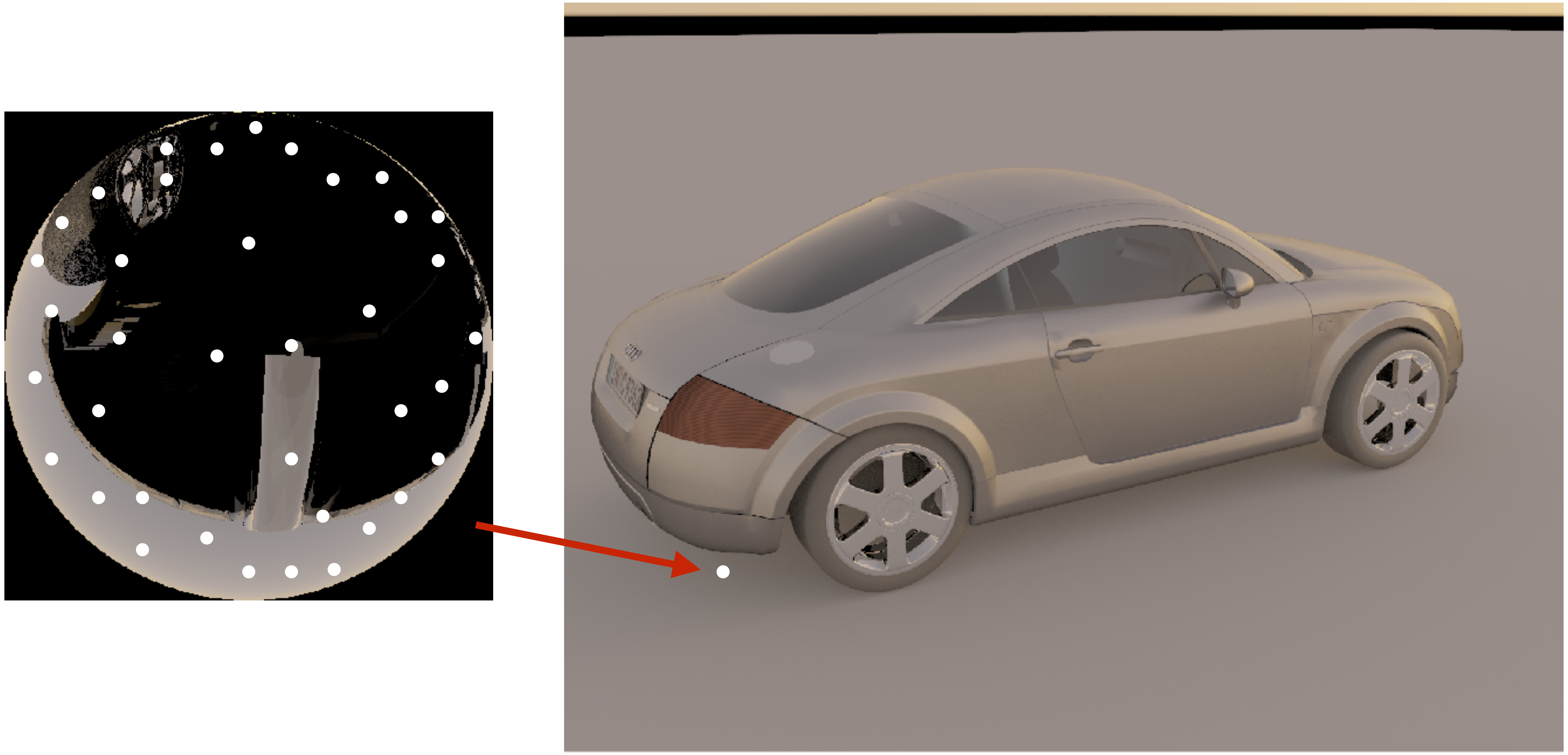
$$\int_0^1 \frac{1}{3}x^2\,dx = \left[x^3\right]_0^1 = 1$$

- For everything else, we have to compute a numerical approximation
- Basic idea:
  - integral is "area under curve"
  - sample the function at many points
  - integral is approximated as weighted sum



Rendering: what are we integrating?

- Recall this view of the world:

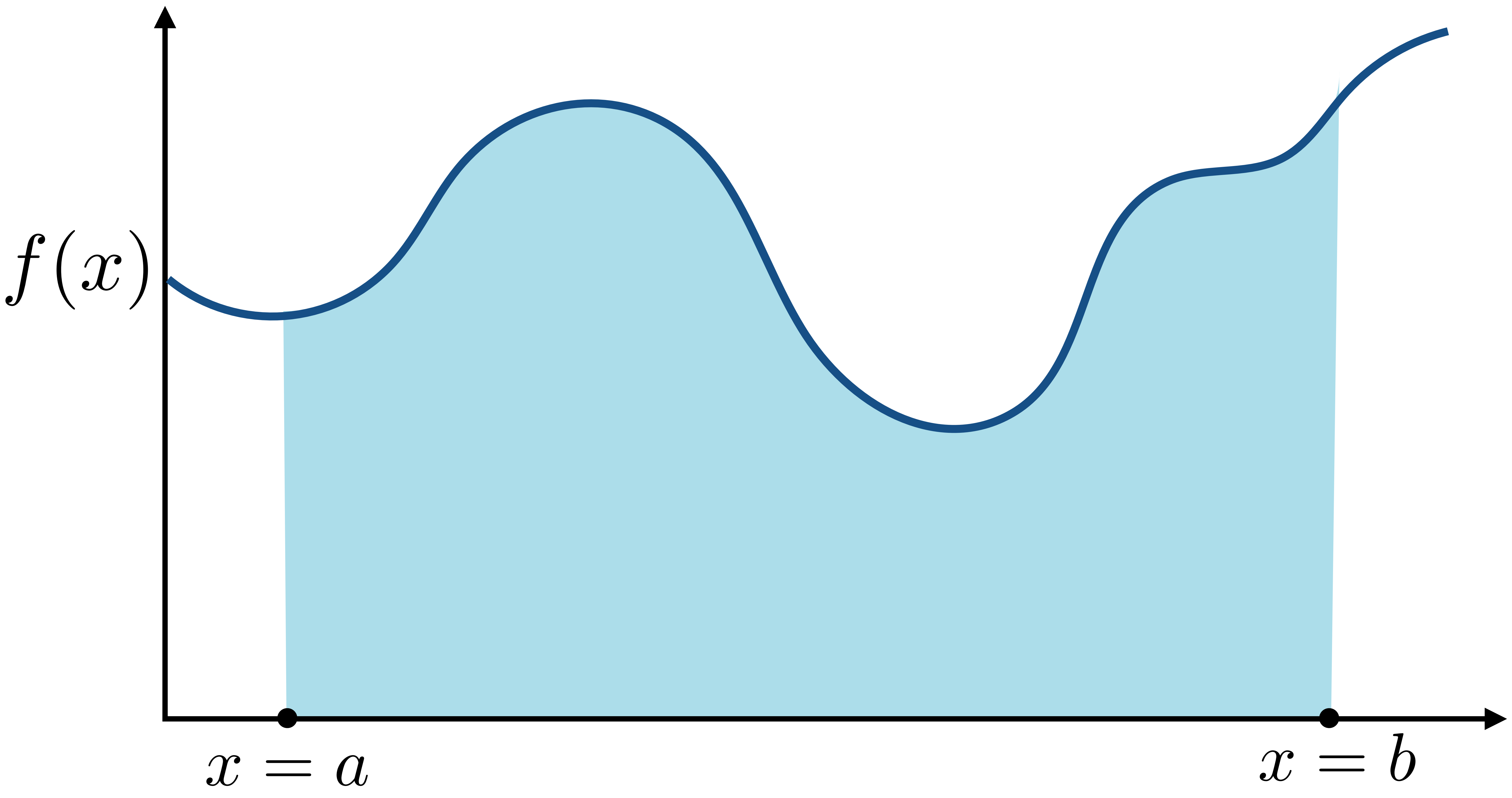


Want to "sum up"—i.e., integrate!—light from all directions

(But let's start a little simpler…)

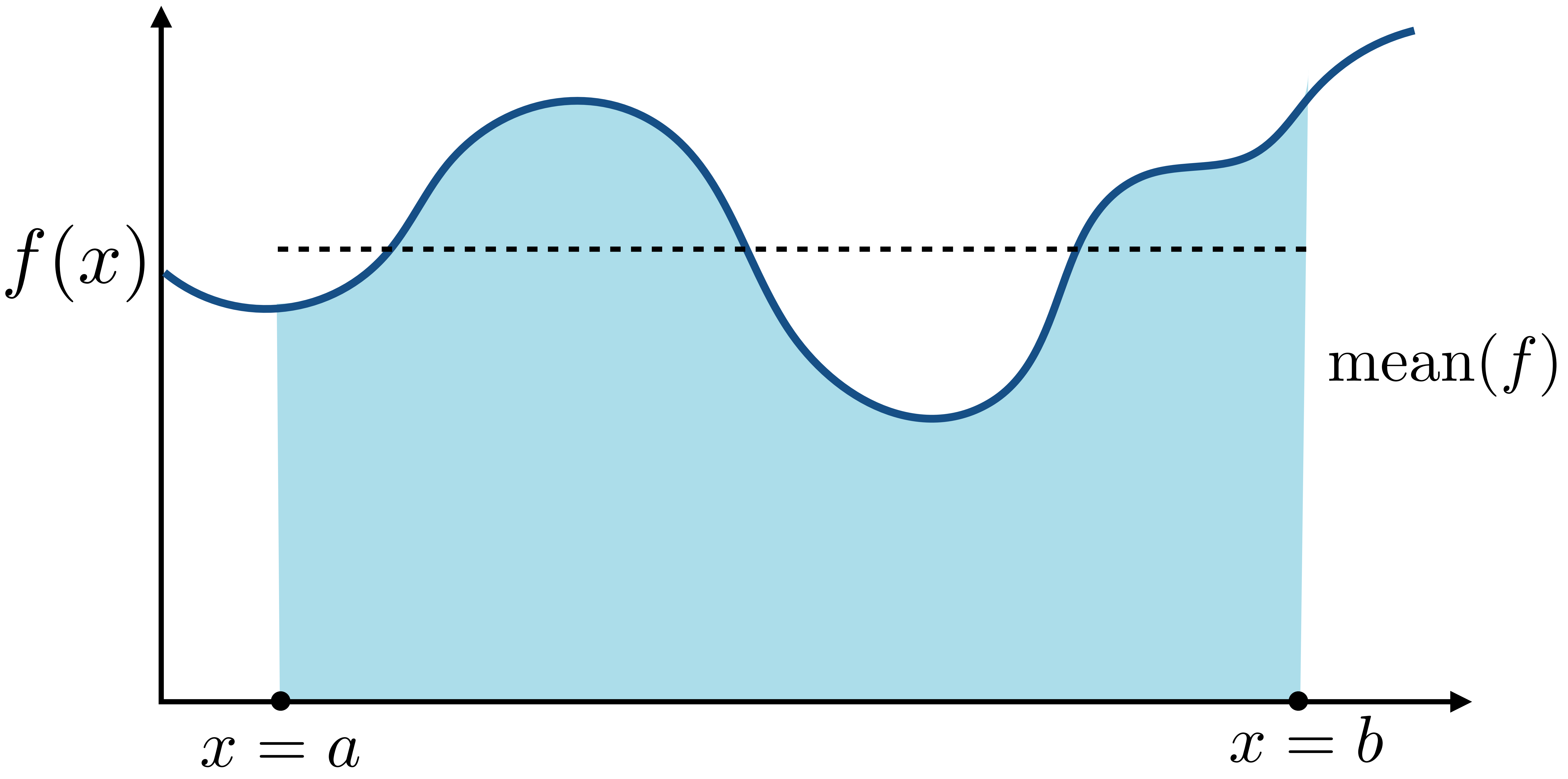
Review: integral as "area under curve"

$$\int_a^b f(x)dx$$



Or: average value times size of domain

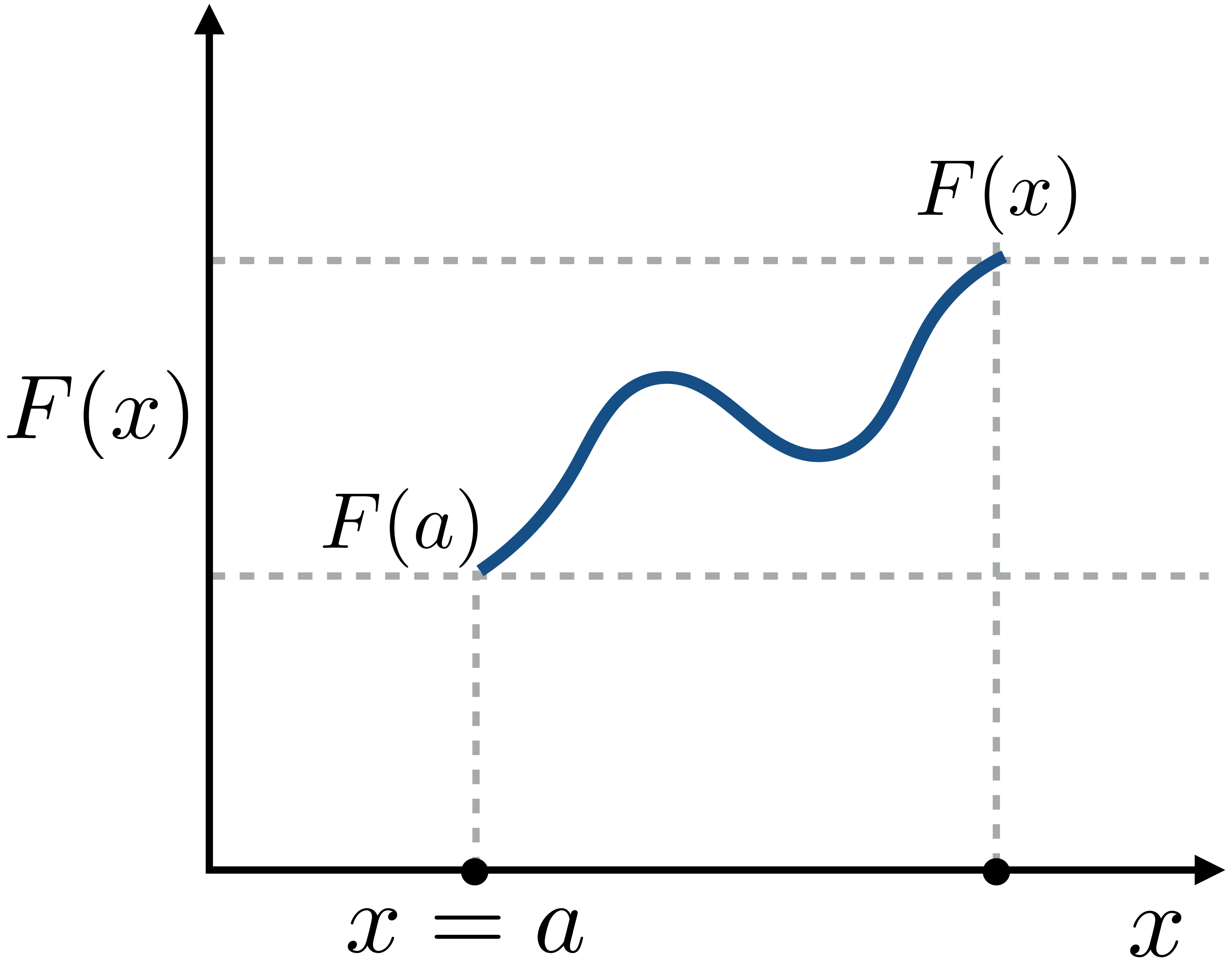
$$\int_a^b f(x)dx = (b-a)\text{mean}(f)$$



Review: fundamental theorem of calculus

$$\int_a^b f(x)dx = F(b) - F(a)$$

$$f(x) = \frac{d}{dx}F(x)$$

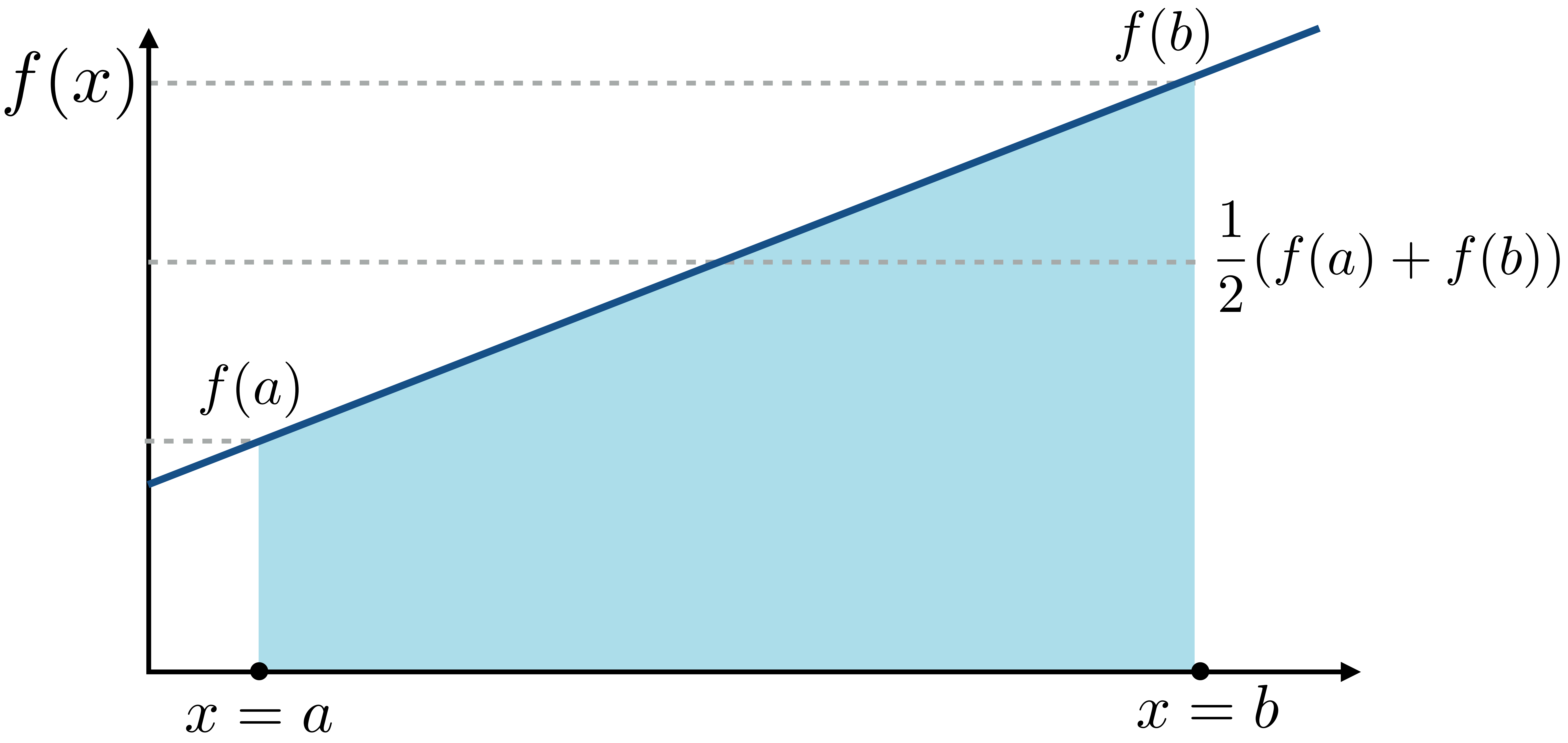


Simple case: constant function

$$\int_a^b C\,dx = (b-a)C$$

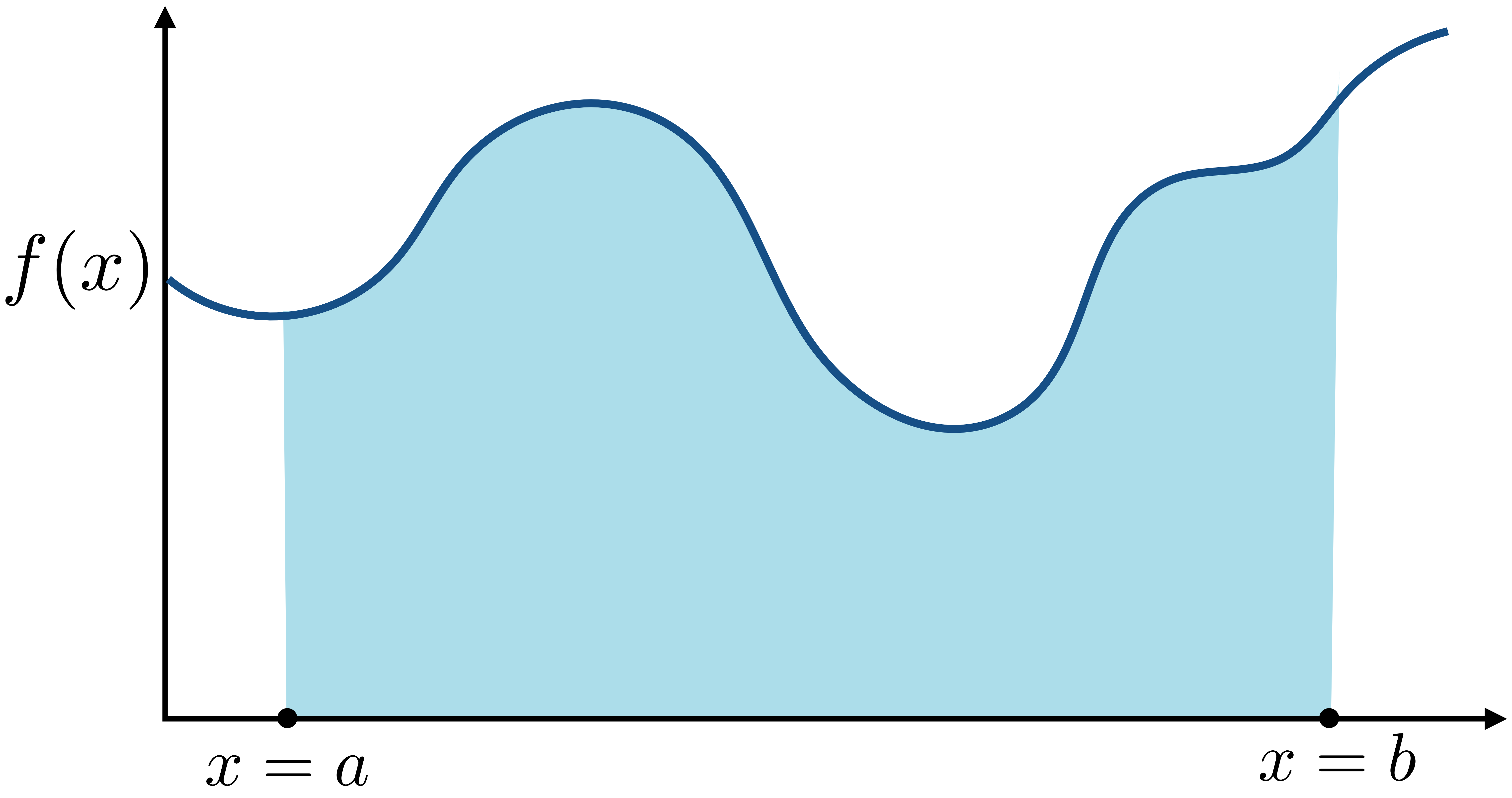
Affine function: $f(x) = cx + d$

$$\int_a^b f(x)dx = \frac{1}{2}(f(a) + f(b))(b - a)$$



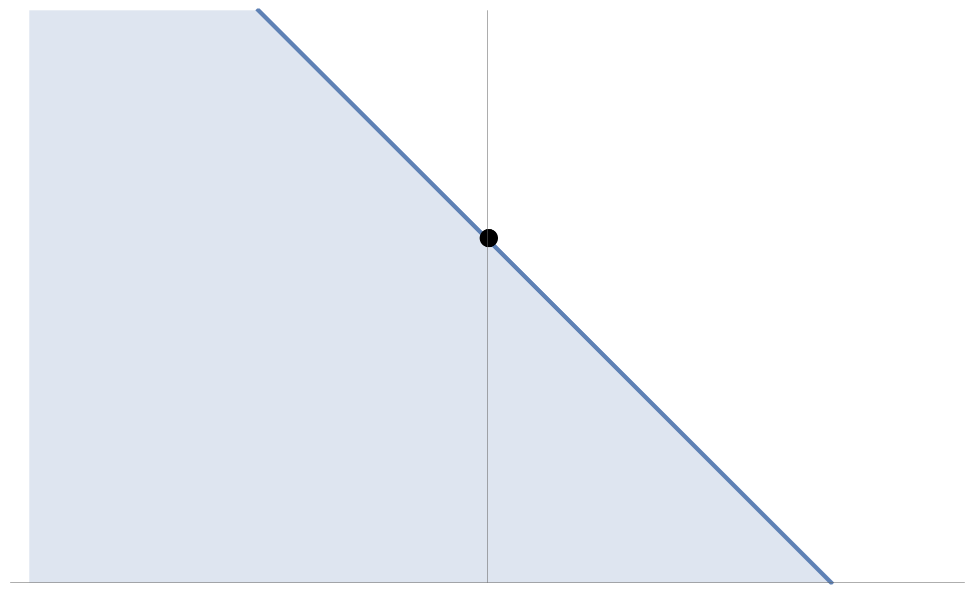
Need only *one* sample of the function (at just the right place…)

More general polynomials?

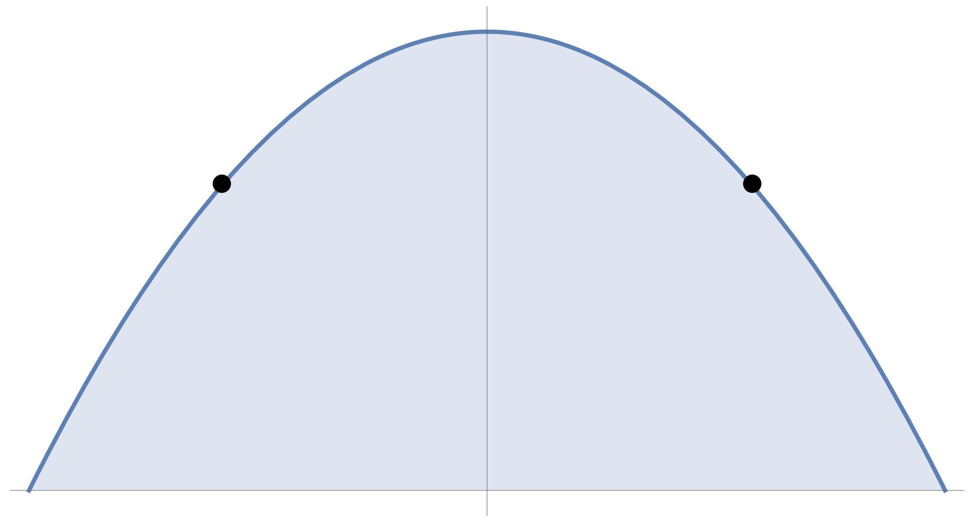


Gauss Quadrature

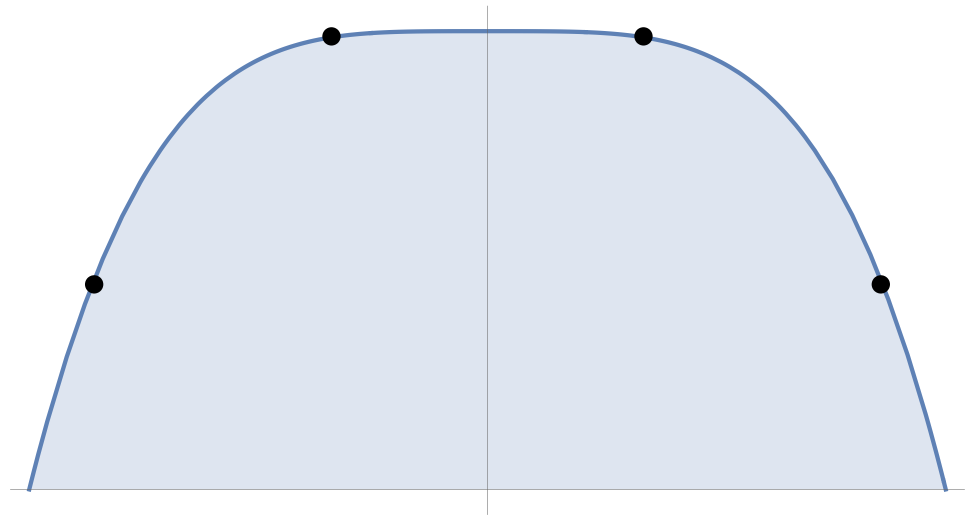
- **For any polynomial of degree *n*, we can always obtain the exact integral by sampling at a special set of *n* points and taking a special weighted combination**



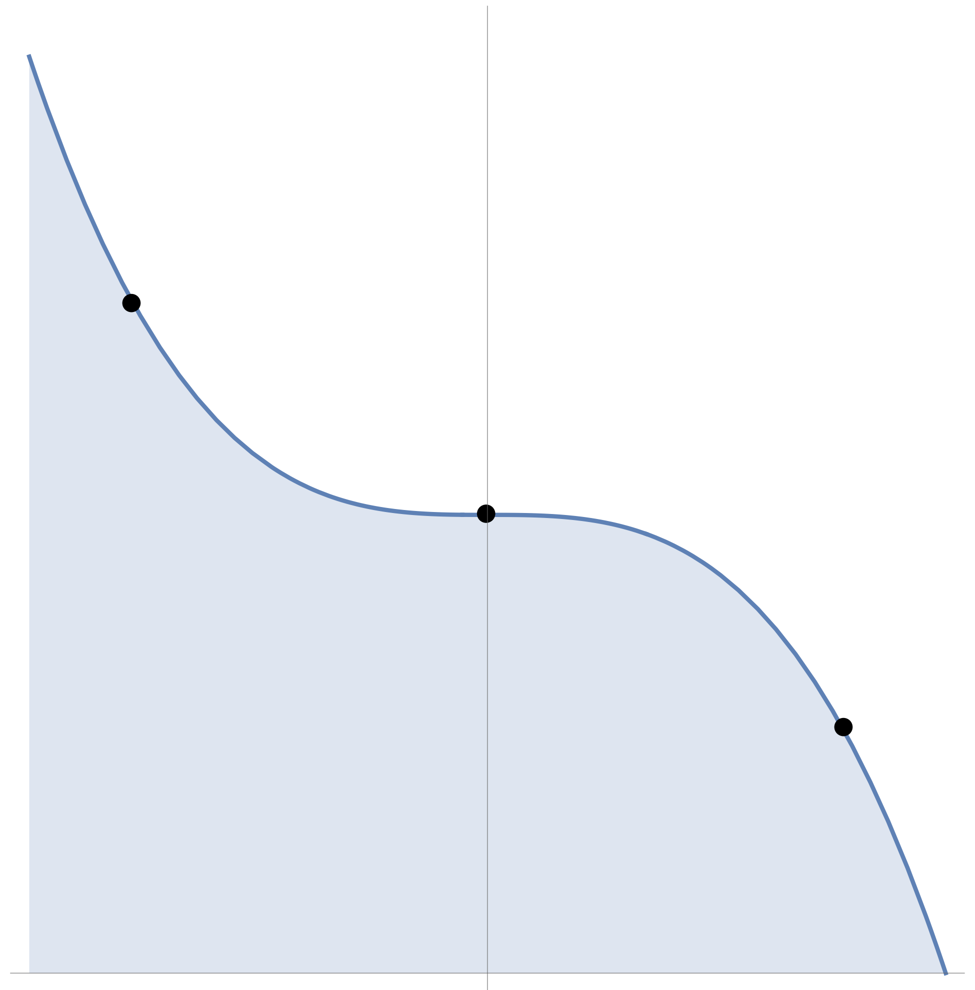
n=1



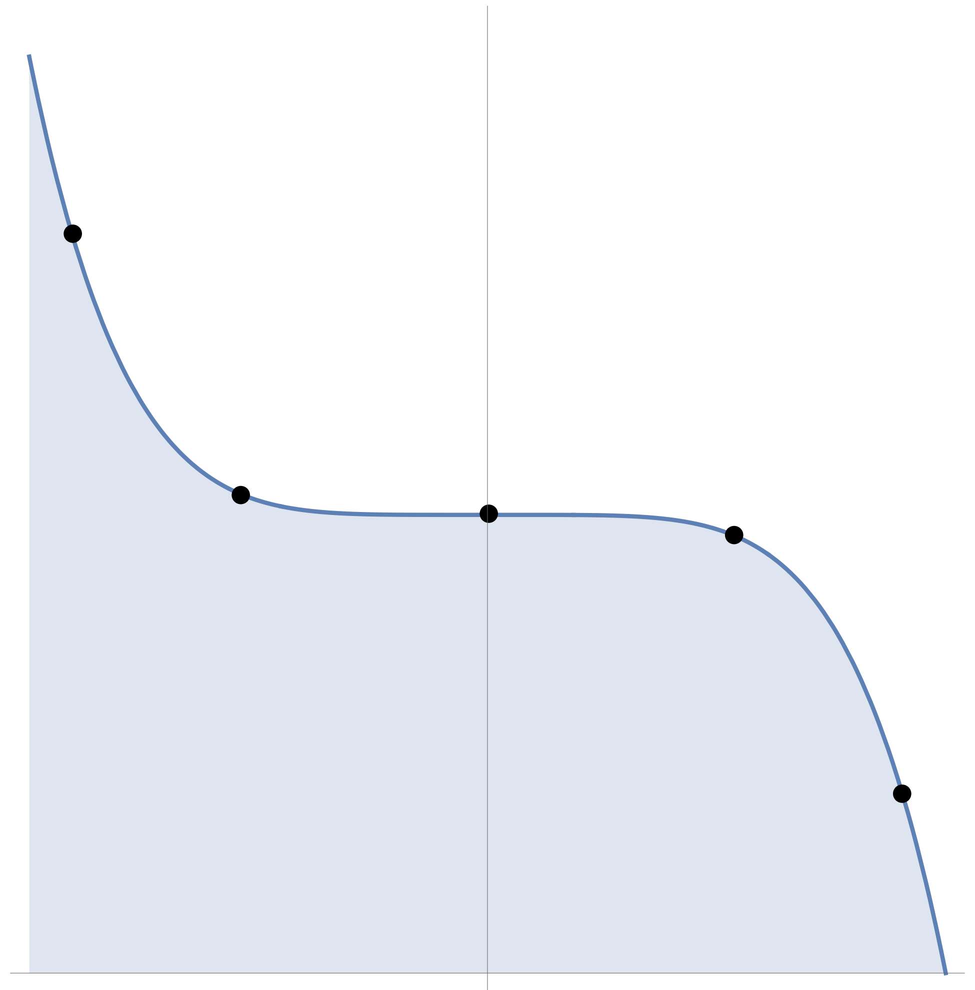
n=2



n=4



n=3

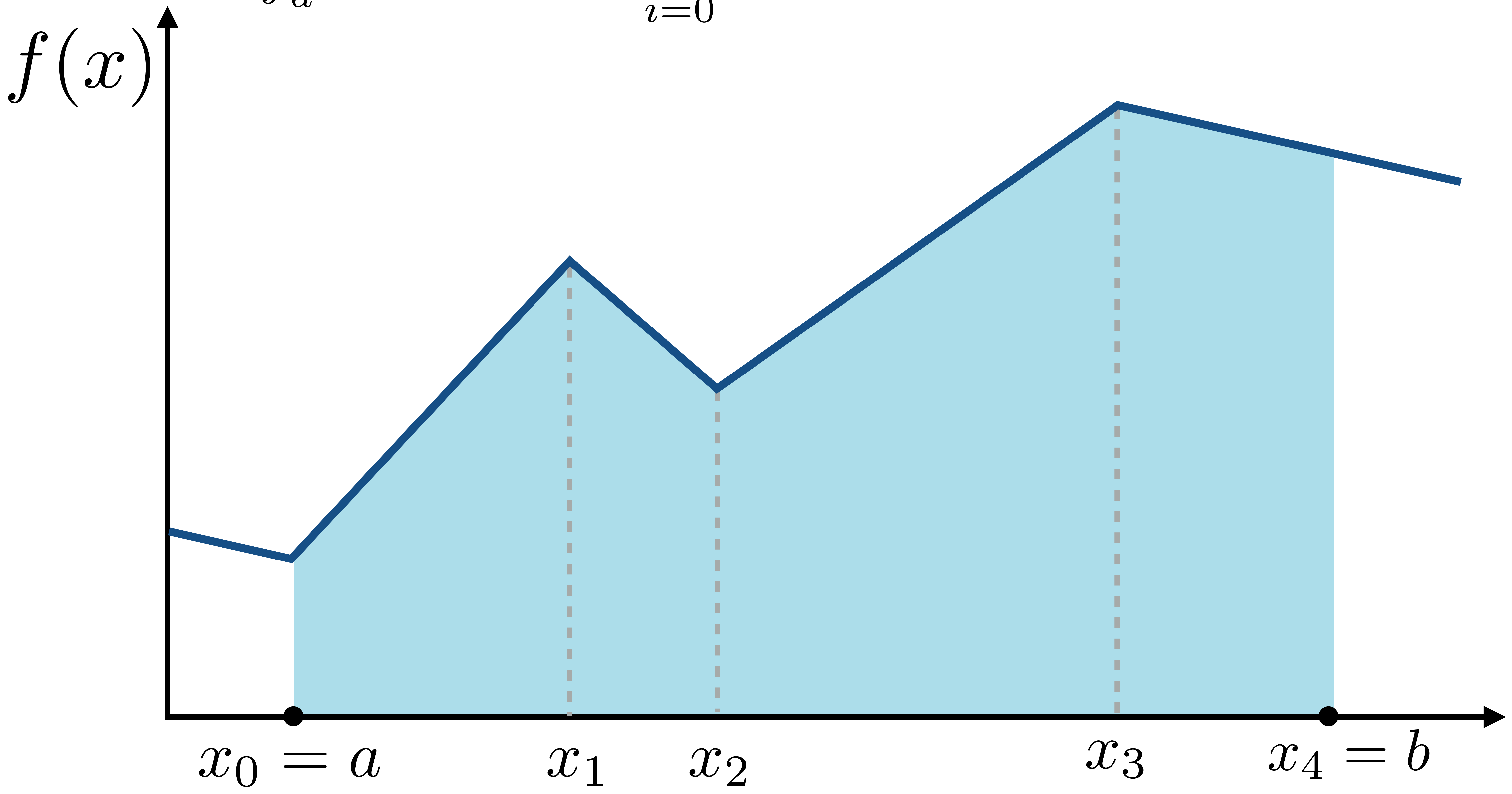


n=5

Piecewise affine function

For piecewise functions, just sum integral of each piece:

$$\int_a^b f(x)dx = \frac{1}{2}\sum_{i=0}^{n-1}(x_{i+1} - x_i)(f(x_i) + f(x_{i+1}))$$



Key idea so far:

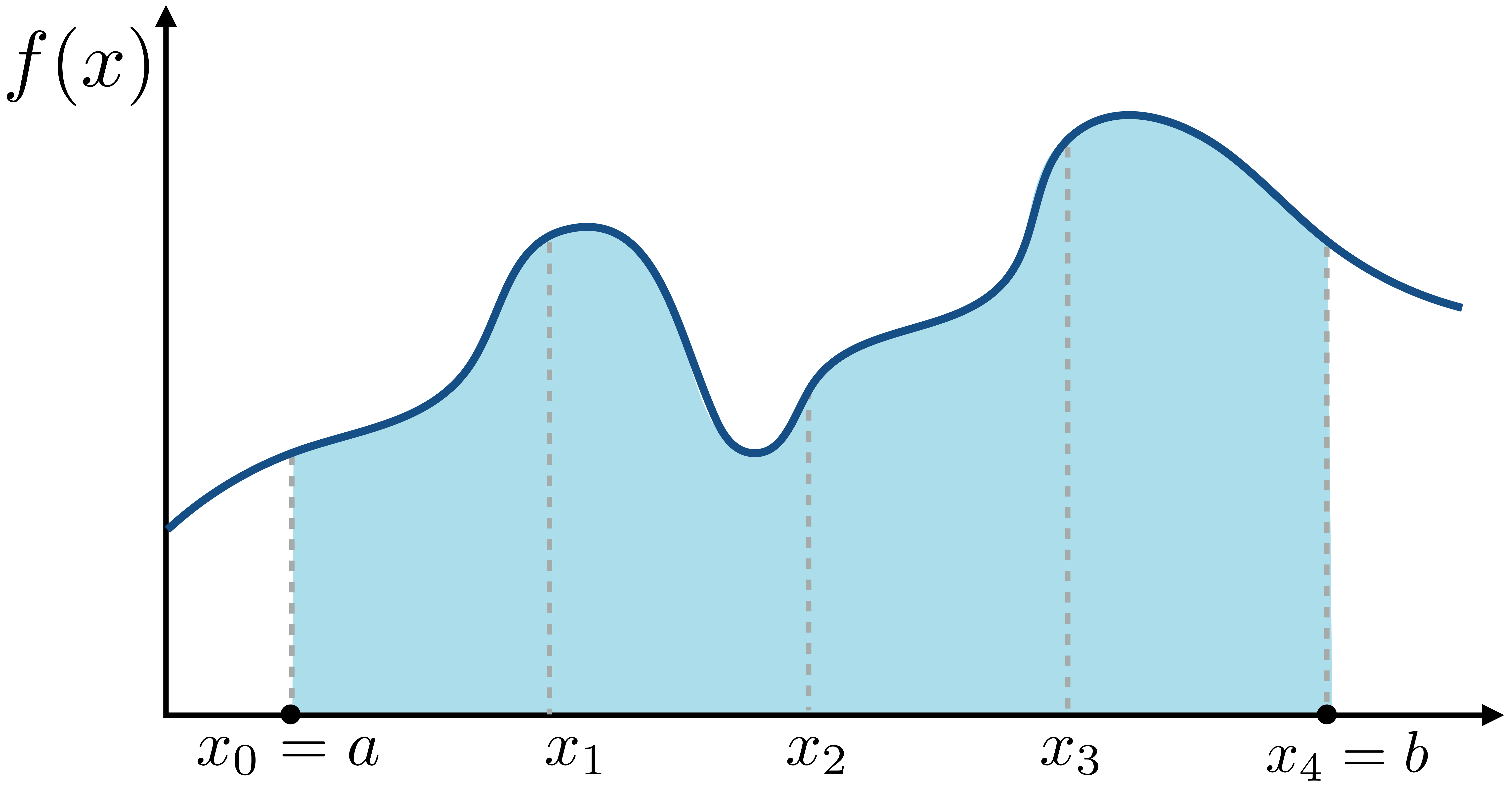
To approximate an integral, we need

(i) quadrature points, and

(ii) weights for each point

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

Arbitrary function f(x)?

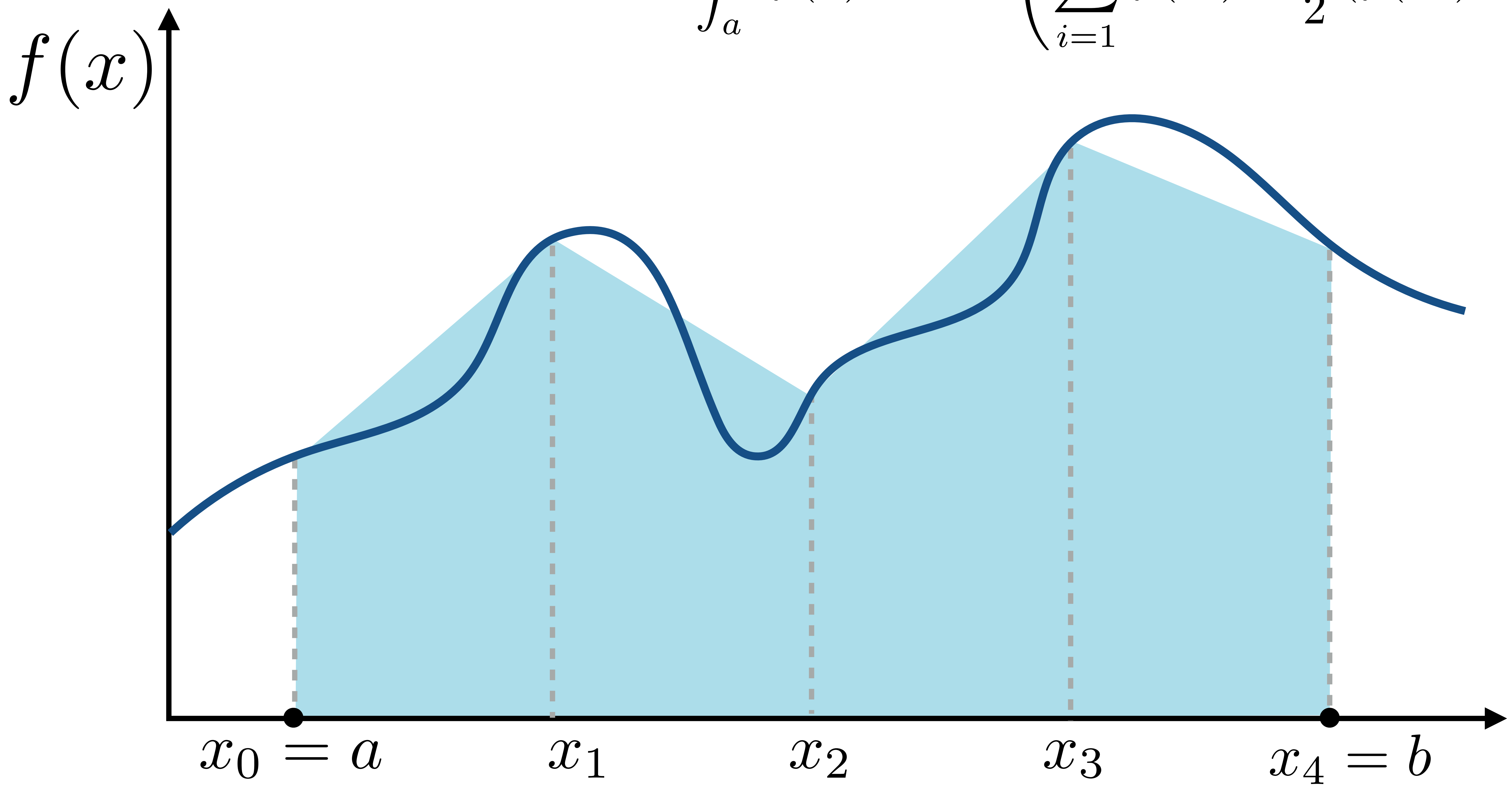


Trapezoid rule

Approximate integral of f(x) by pretending function is piecewise affine

For equal length segments: $h = \frac{b-a}{n-1}$

$$\int_a^b f(x)dx = h\left(\sum_{i=1}^{n-1} f(x_i) + \frac{1}{2}\left(f(x_0) + f(x_n)\right)\right)$$



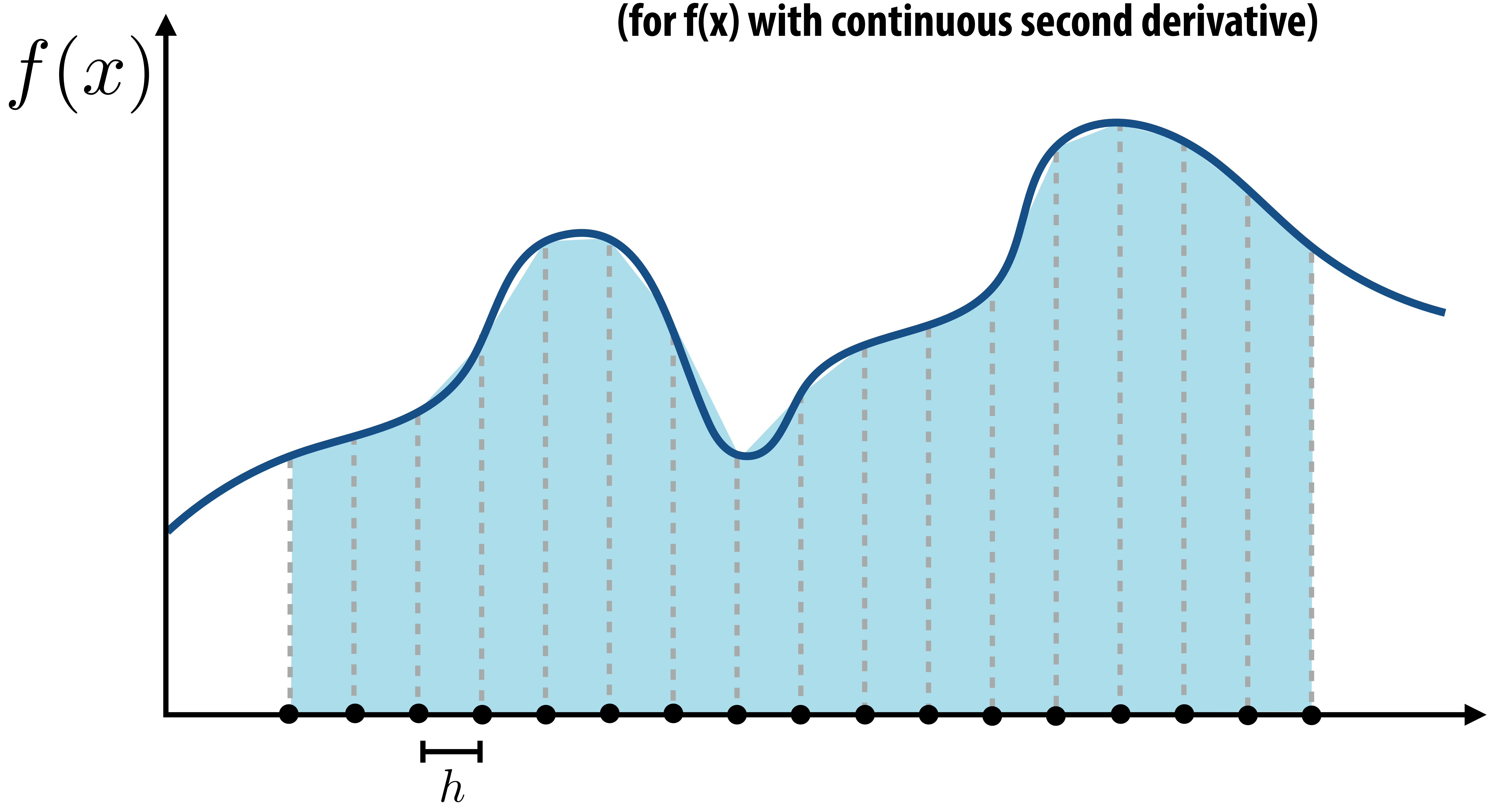
Trapezoid rule

Consider cost and accuracy of estimate as $n \to \infty$ (or $h \to 0$)

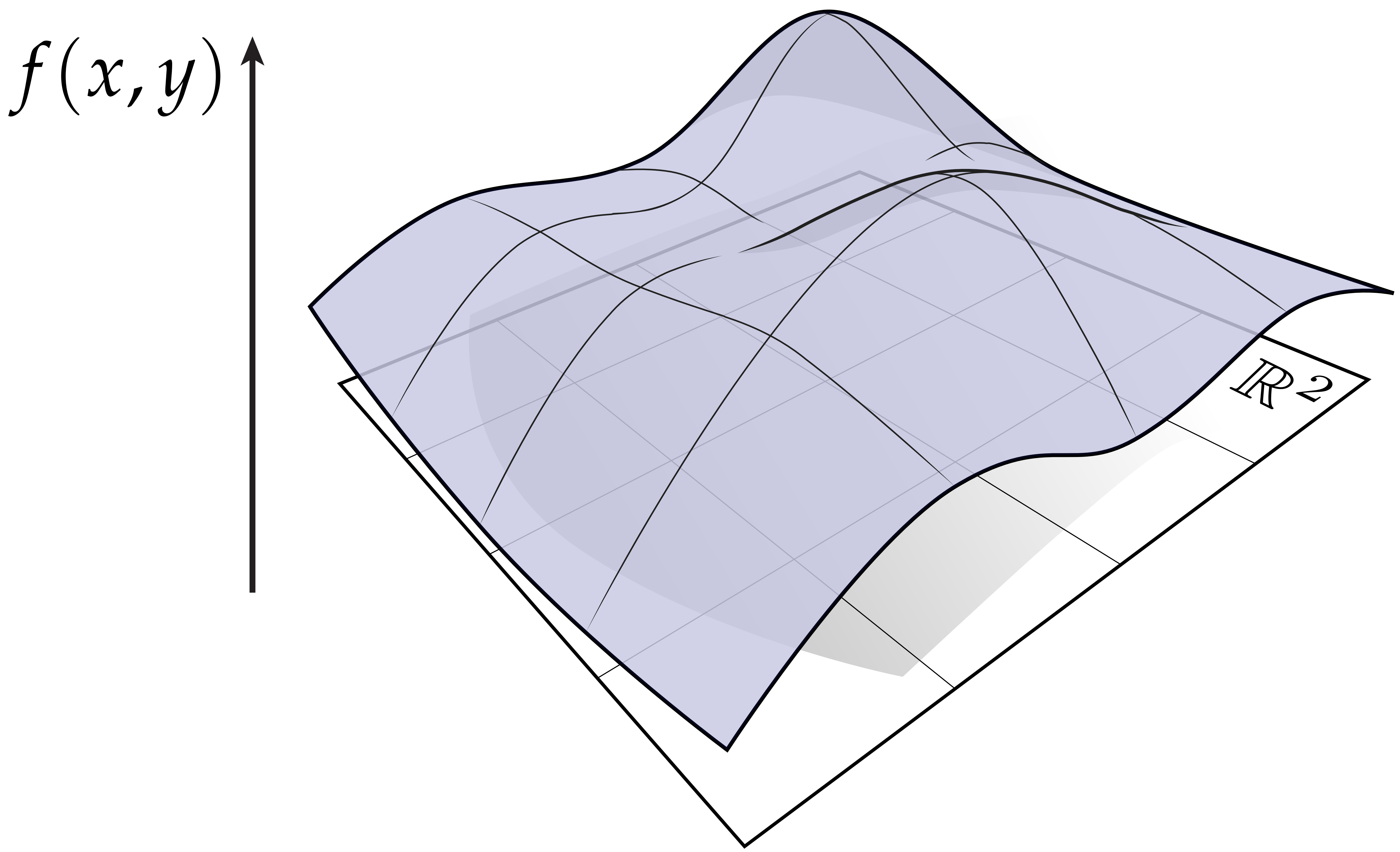
Work: $O(n)$

Error can be shown to be: $O(h^2) = O(\frac{1}{n^2})$

(for f(x) with continuous second derivative)



What about a 2D function?



How should we approximate the area underneath?

Integration in 2D

Consider integrating $f(x,y)$ **using the trapezoidal rule (apply rule twice: when integrating in *x* and in *y*)**

$$\int_{a_y}^{b_y}\int_{a_x}^{b_x} f(x,y)dxdy = \int_{a_y}^{b_y}\left(O(h^2)+\sum_{i=0}^{n}A_i f(x_i,y)\right)dy$$

First application of rule

$$= O(h^2)+\sum_{i=0}^{n}A_i\int_{a_y}^{b_y} f(x_i,y)dy$$

$$= O(h^2)+\sum_{i=0}^{n}A_i\left(O(h^2)+\sum_{j=0}^{n}A_j f(x_i,y_j)\right)$$

Second application

$$= O(h^2)+\sum_{i=0}^{n}\sum_{j=0}^{n}A_iA_j f(x_i,y_j)$$

Errors add, so error still: $O(h^2)$

But work is now: $O(n^2)$

(*n* x *n* set of measurements)

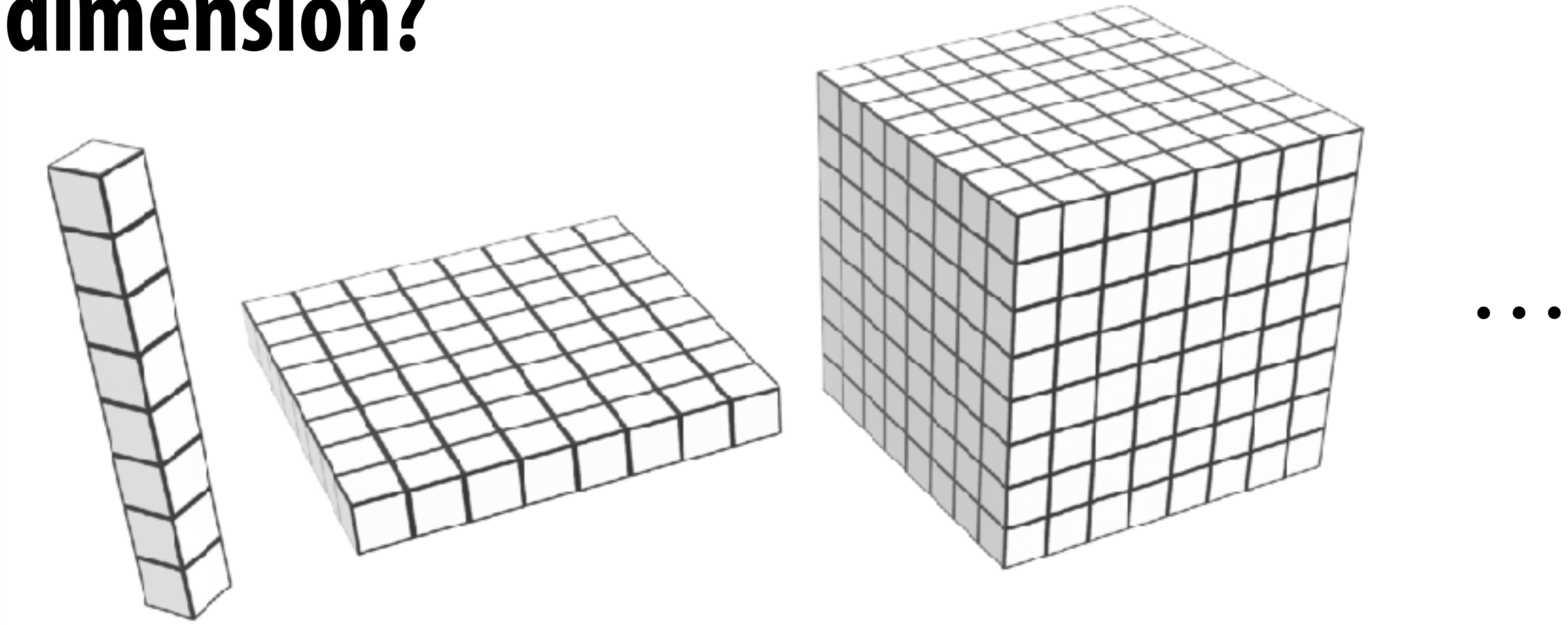
Must perform much more work in 2D to get same error bound on integral!

In K-D, let $N = n^k$

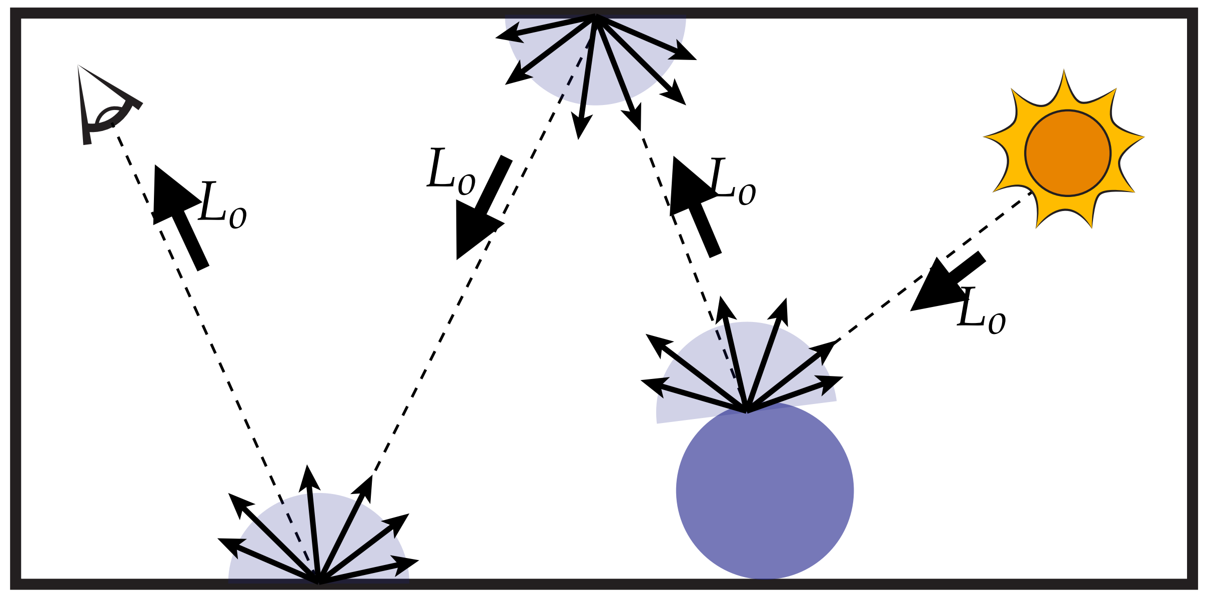
Error goes as: $O\left(\frac{1}{N^{2/k}}\right)$

Curse of Dimensionality

- How much does it cost to apply the trapezoid rule as we go up in dimension?
  - 1D: O(n)
  - 2D: O(n^2)
  - …
  - kD: O(n^k)



- For many problems in graphics (like rendering), *k* is very, very big (e.g., tens or hundreds or thousands)



- Applying trapezoid rule does not scale!
- Need a fundamentally different approach…

Monte Carlo Integration

Credit: many of these slides were created by Matt Pharr and Pat Hanrahan

Monte Carlo Integration

So far we've discussed techniques that use a fixed set of sample points (e.g., uniformly spaced, or obtained by finding roots of polynomial (Gaussian quadrature))

- Estimate value of integral using random sampling of function
  - Value of estimate depends on random samples used
  - But algorithm gives the correct value of integral "on average"
- Only requires function to be evaluated at random points on its domain
  - Applicable to functions with discontinuities, functions that are impossible to integrate directly
- Error of estimate is independent of the dimensionality of the integrand
  - Depends on the number of random samples used: $O(n^{1/2})$

Review: random variables

X **random variable. Represents a distribution of potential values**

$X \sim p(x)$ **probability density function (PDF). Describes relative probability of a random process choosing value x**

Uniform PDF: all values over a domain are equally likely

e.g., for an unbiased die

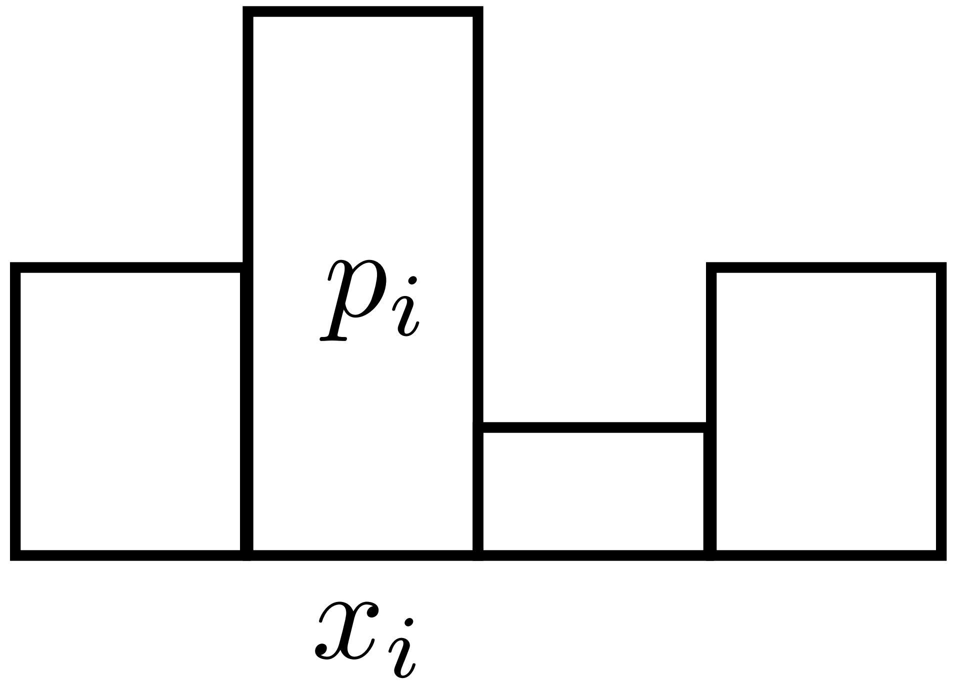
X **takes on values 1,2,3,4,5,6**

$p(1) = p(2) = p(3) = p(4) = p(5) = p(6)$

Discrete probability distributions

n **discrete values** x_i

With probability p_i



Requirements of a PDF:

$$p_i \geq 0$$

$$\sum_{i=1}^{n} p_i = 1$$

Six-sided die example: $p_i = \frac{1}{6}$

Think: p_i **is the probability that a random measurement of** X **will yield the value** x_i

X **takes on the value** x_i **with probability** p_i

Cumulative distribution function (CDF)

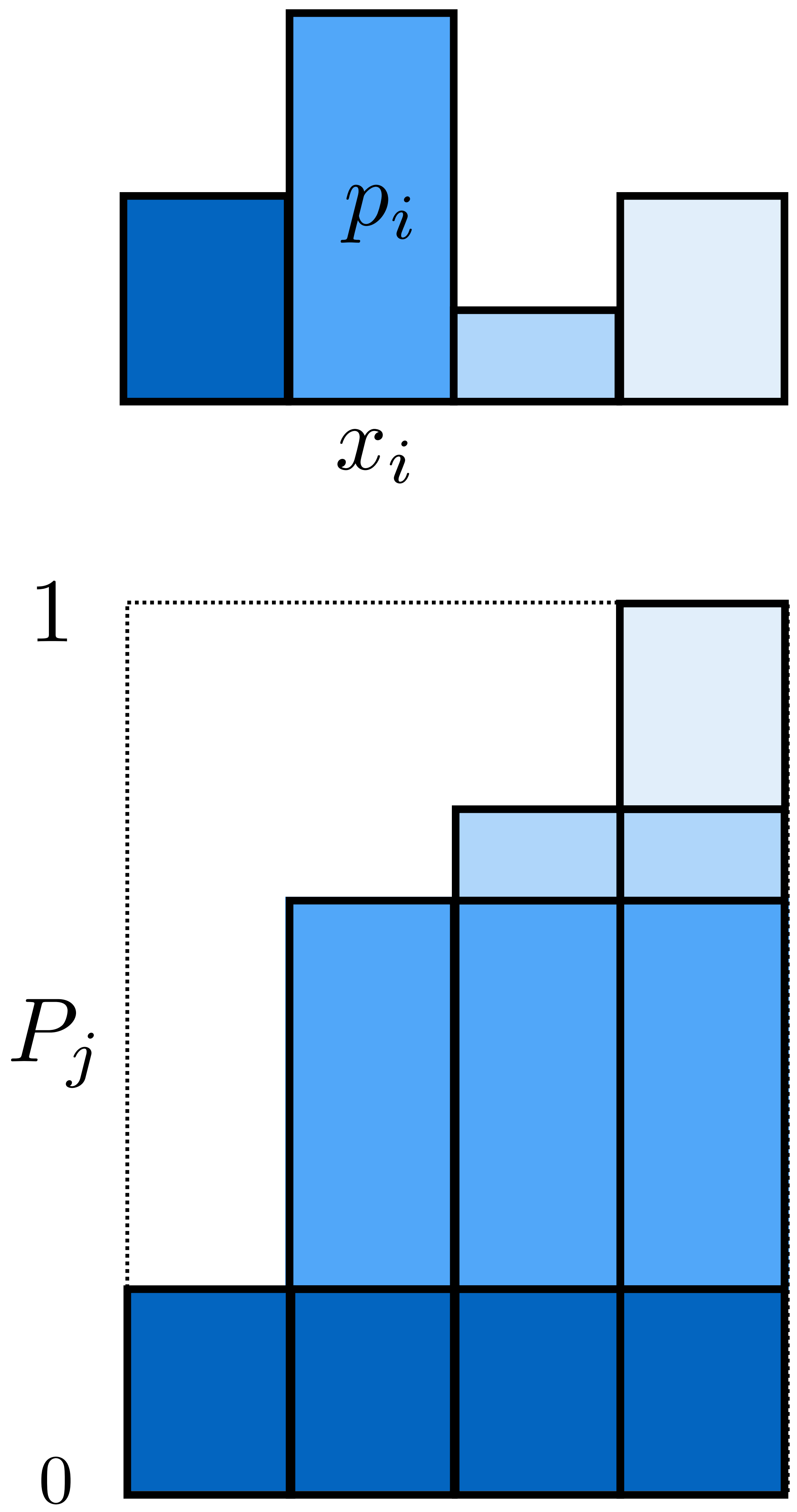
(For a discrete probability distribution)

Cumulative PDF: $P_j = \sum_{i=1}^{j} p_i$

where:

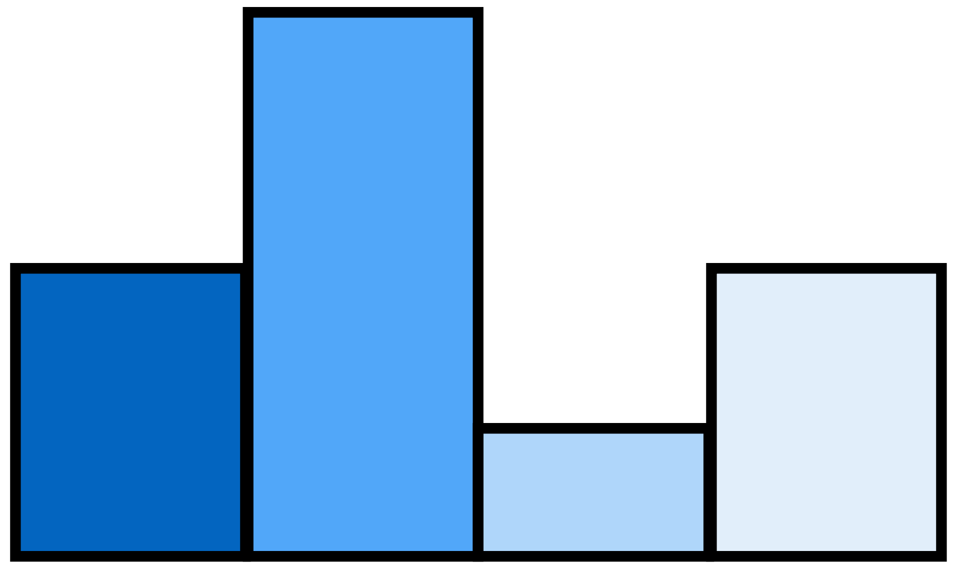
$$0 \leq P_i \leq 1$$

$$P_n = 1$$



How do we generate samples of a discrete random variable (with a known PDF?)

Sampling from discrete probability distributions

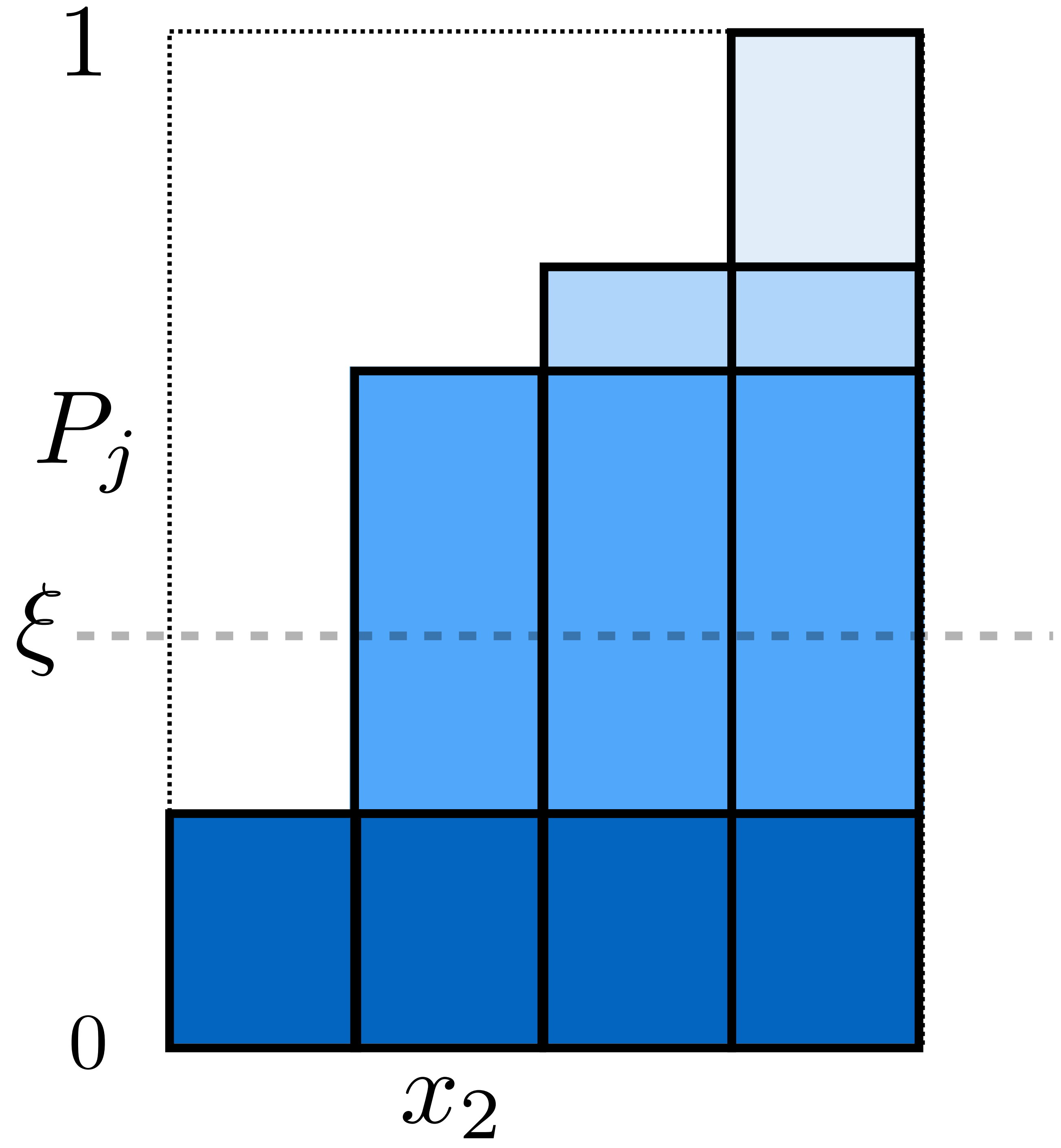


To randomly select an event,
select x_i if

$$P_{i-1} < \xi \leq P_i$$

↑

Uniform random variable $\in [0, 1)$



Note: "inversion method" is O(log N) work per sample. But O(1) is possible! (More on the "Alias Method" later.)

Continuous probability distributions

PDF $p(x)$

$$p(x) \geq 0$$

CDF $P(x)$

$$P(x) = \int_0^x p(x)\,\mathrm{d}x$$

$$P(x) = \Pr(X < x)$$

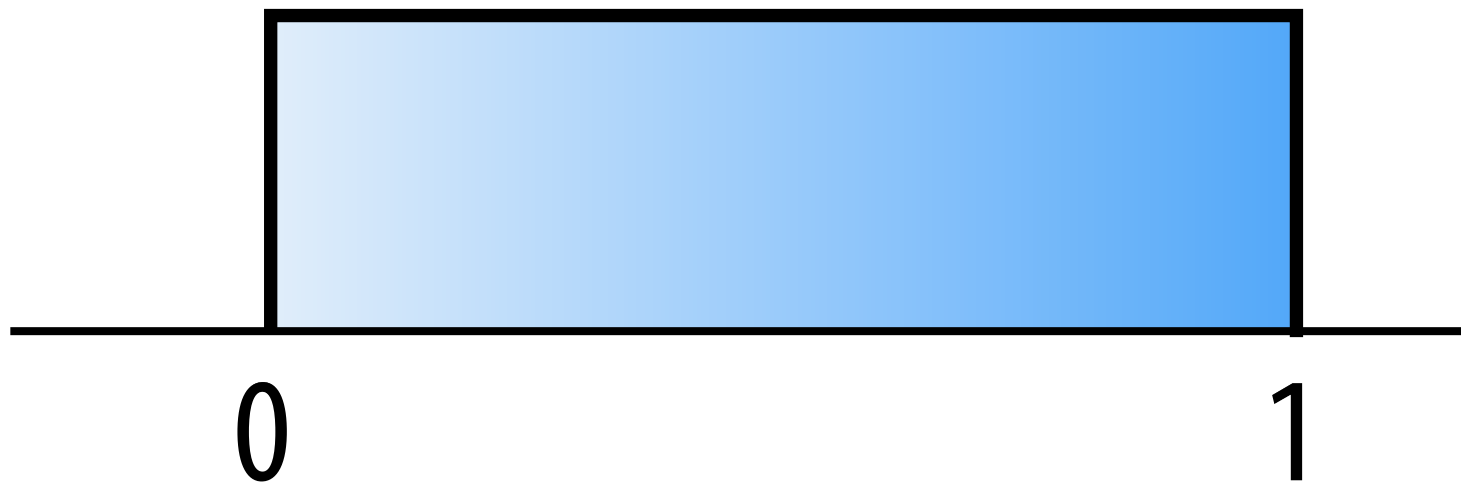
$$P(1) = 1$$

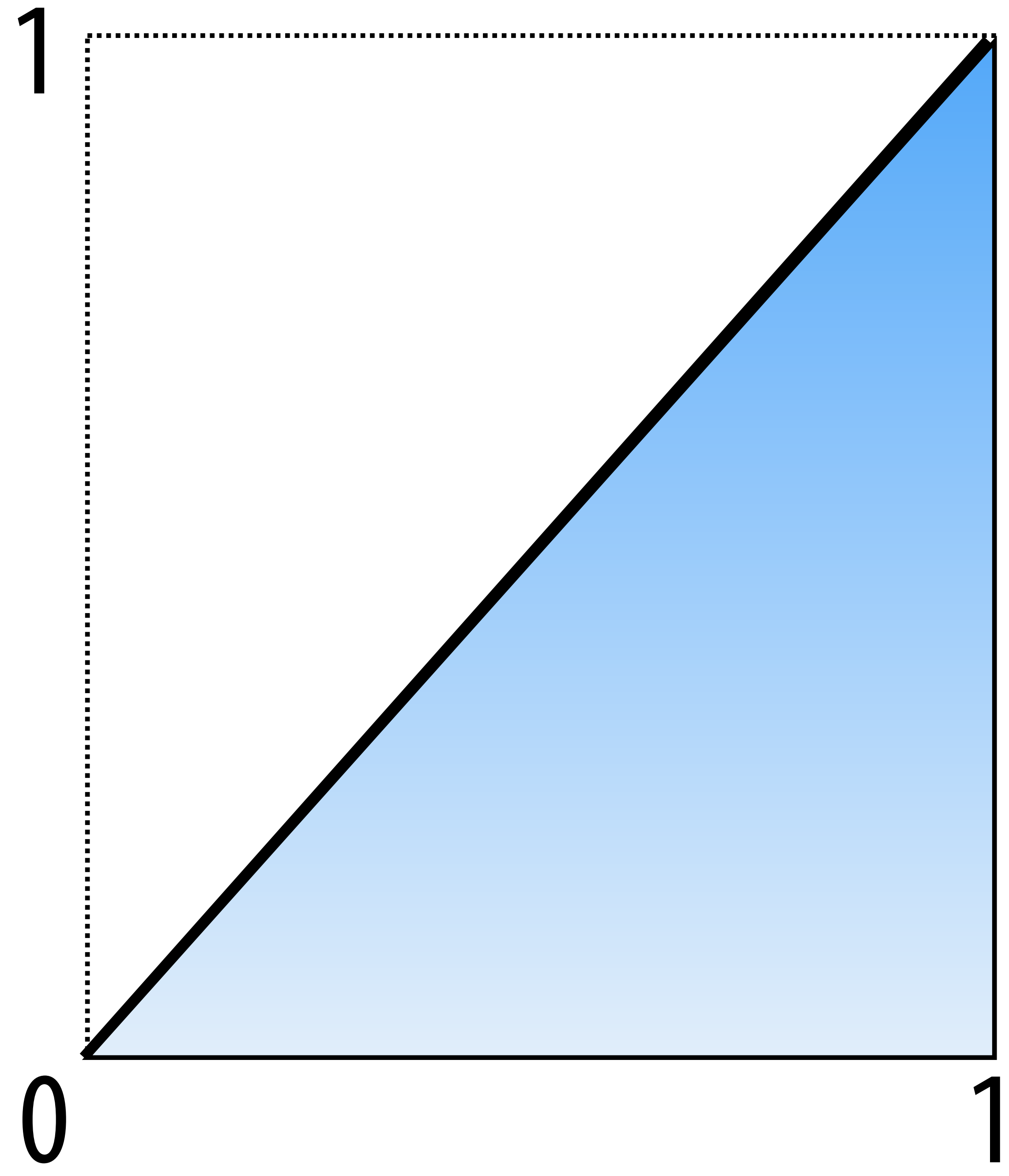
$$\Pr(a \leq X \leq b) = \int_a^b p(x)\,\mathrm{d}x$$

$$= P(b) - P(a)$$

Uniform distribution

(for random variable X defined on [0,1] domain)

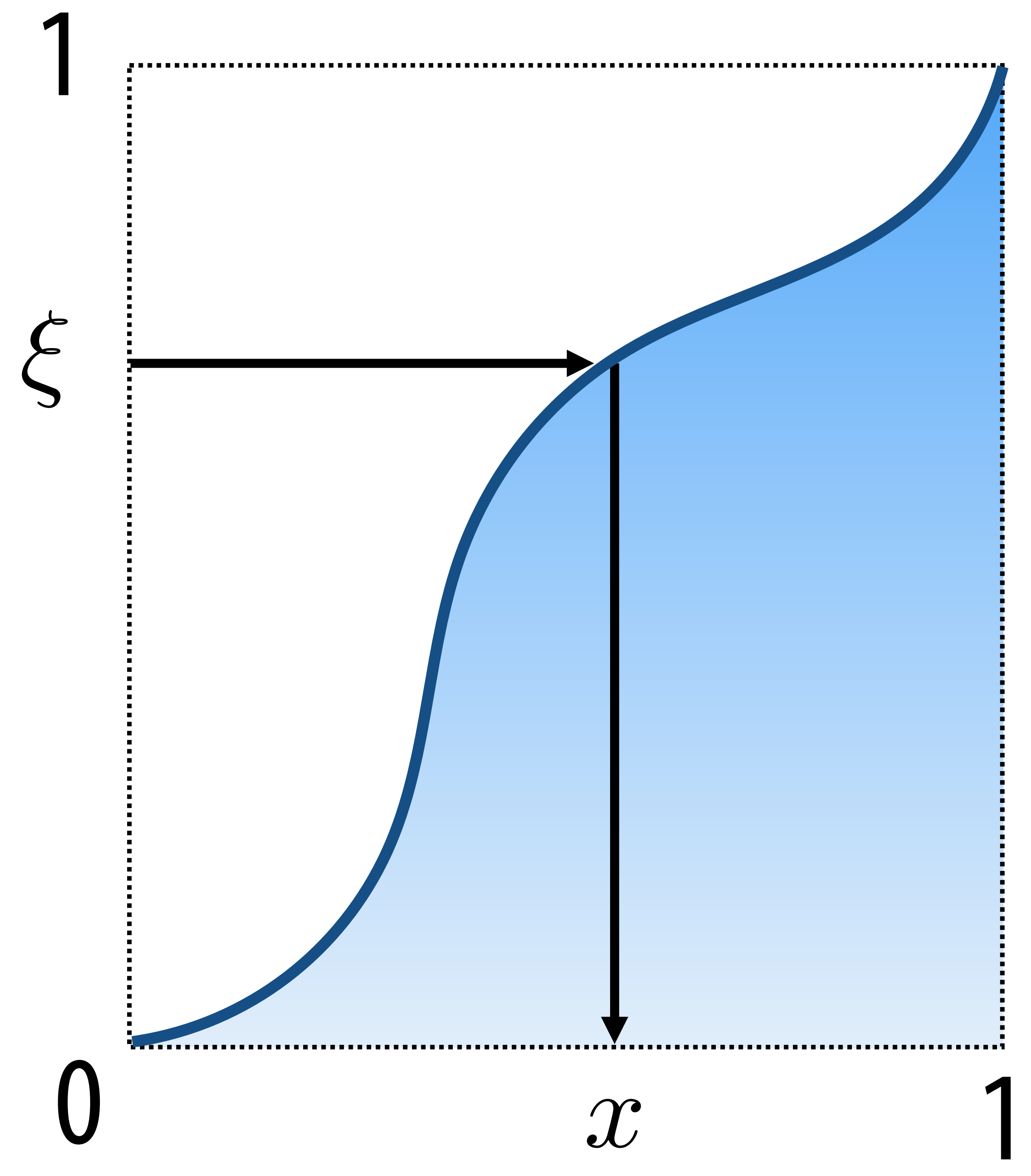




Sampling continuous random variables using the inversion method

Cumulative probability distribution function

$$P(x) = \Pr(X < x)$$



Construction of samples:

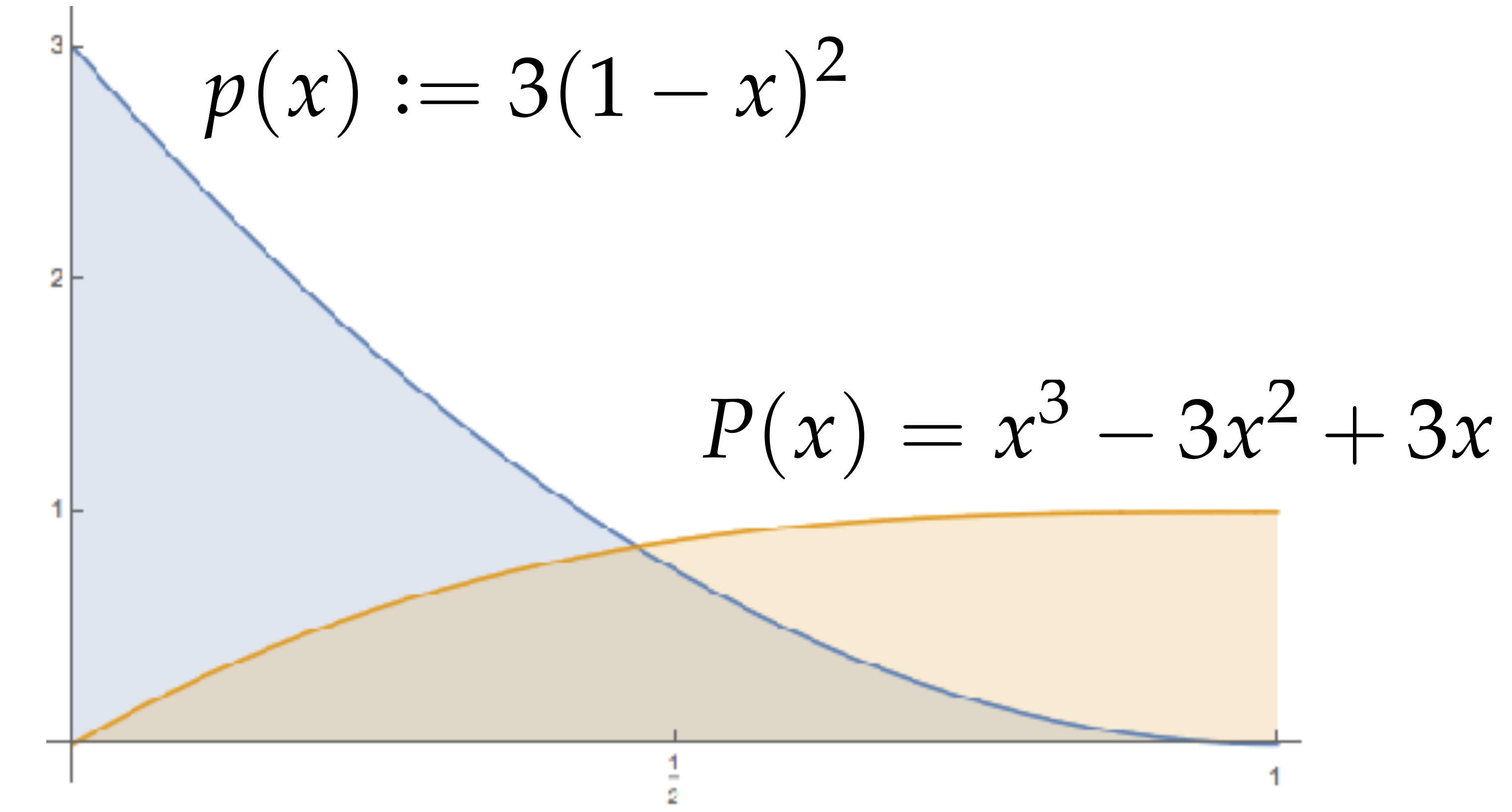
Solve for $x = P^{-1}(\xi)$

Must know the formula for:

1. **The integral of** $p(x)$
2. **The inverse function** $P^{-1}(x)$

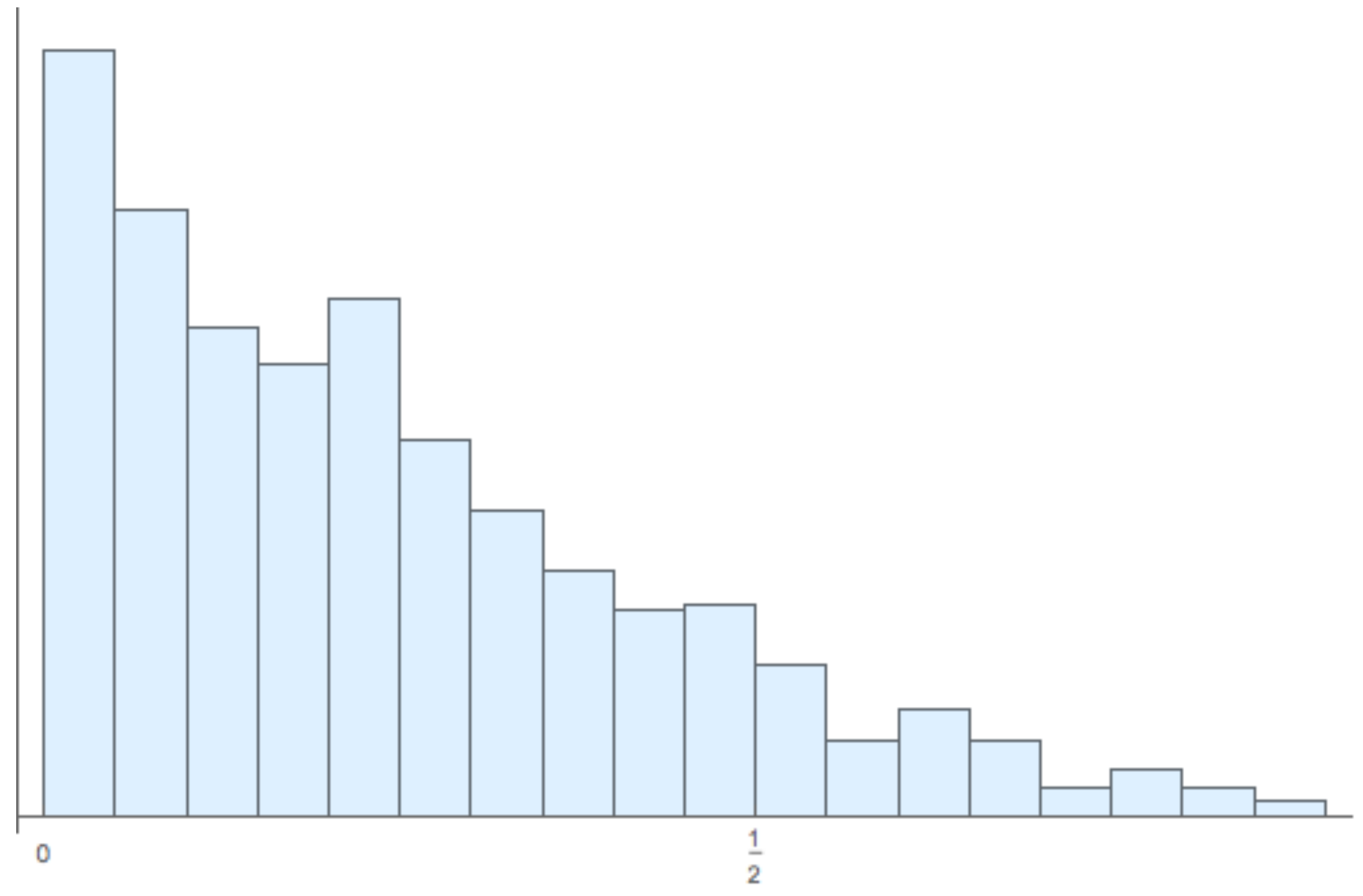
Example—Sampling Quadratic Distribution

- **As a toy example, consider the simple probability distribution p(x) := 3(1-x)² over the interval [0,1]**
- **How do we pick random samples distributed according to p(x)?**
- **First, integrate probability distribution p(x) to get cumulative distribution P(x)**
- **Invert P(x) by solving y = P(x) for x**
- **Finally, plug uniformly distributed random values y in [0,1] into this expression**

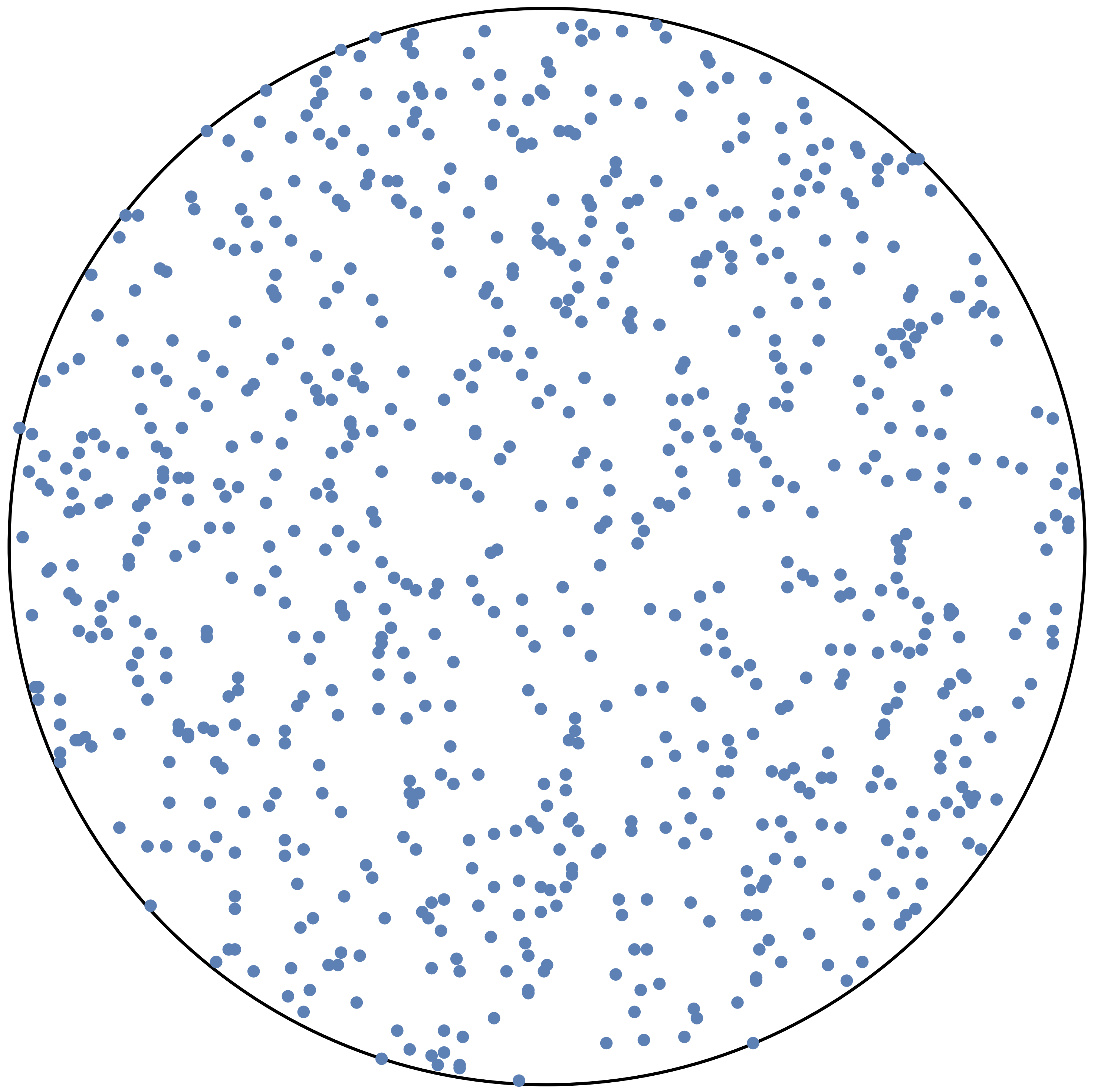


$$\int_0^s 3(1-x)^2\,dx = s^3 - 3s^2 + 3s$$

$$x = 1 - (1-y)^{\frac{1}{3}}$$



How do we uniformly sample the unit circle?



I.e., choose any point P=(px, py) in circle with equal probability)

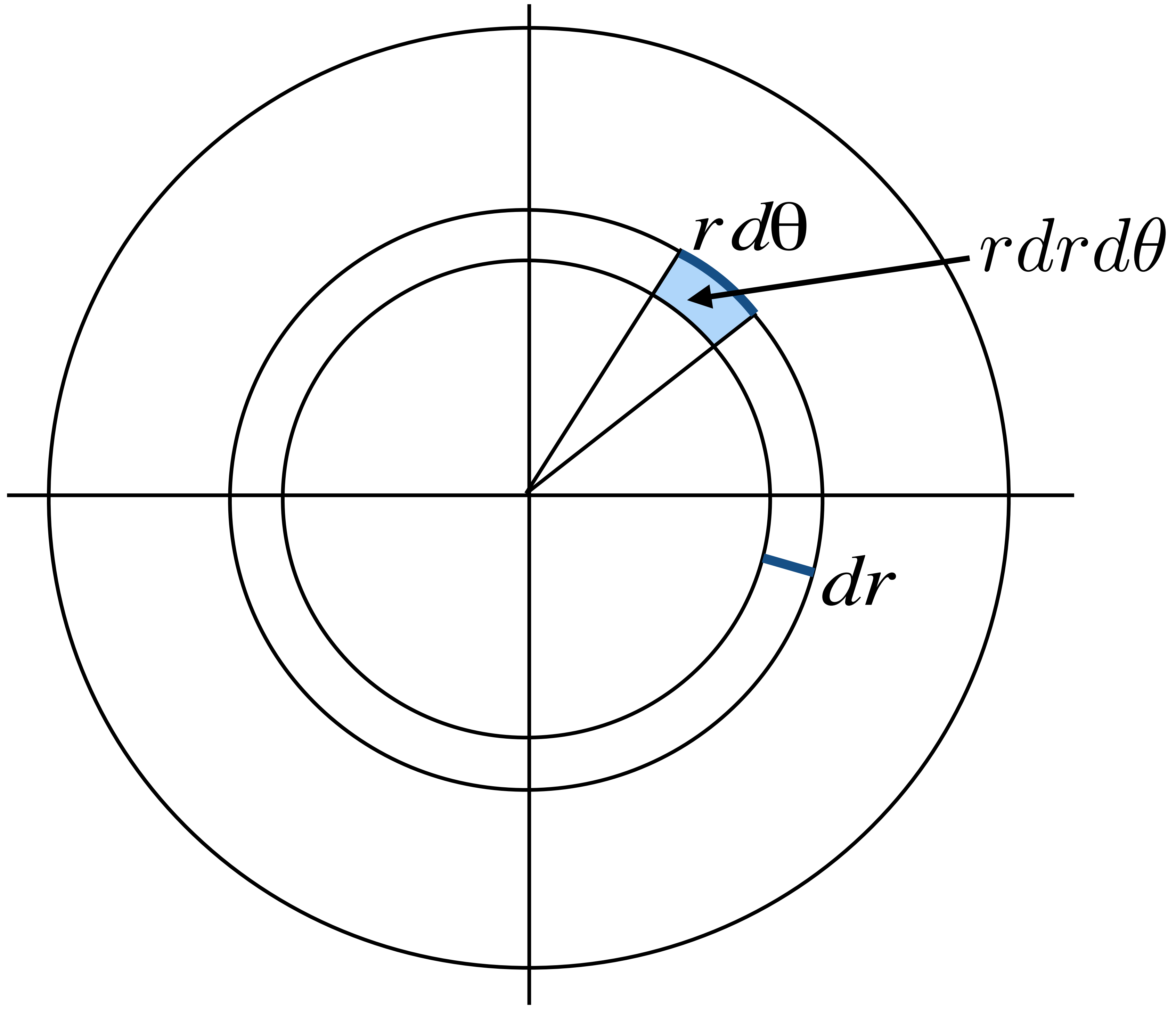
Uniformly sampling unit circle: first try

- θ = uniform random angle between 0 and 2π
- r = uniform random radius between 0 and 1
- Return point: $(r\cos\theta, r\sin\theta)$

This algorithm does not produce the desired uniform sampling of the area of a circle. Why?

Because sampling is not uniform in area!

Points farther from center of circle are less likely to be chosen



$$\theta = 2\pi\xi_1 \qquad r = \xi_2$$

So how should we pick samples? Well, think about how we integrate over a disk in polar coordinates…

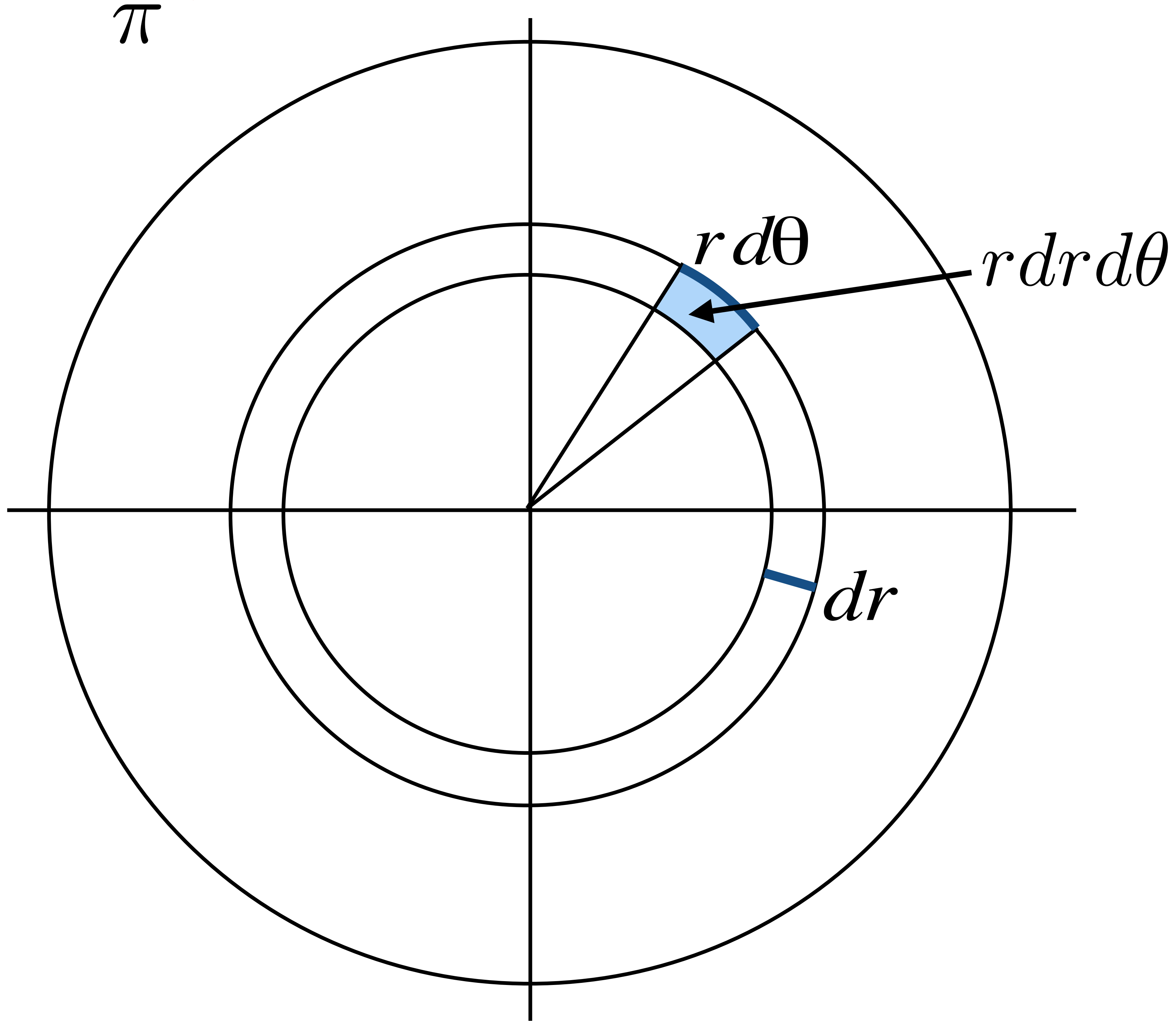
Sampling a circle (via inversion in 2D)

$$A = \int_0^{2\pi}\int_0^1 r\,\mathrm{d}r\,\mathrm{d}\theta = \int_0^1 r\,\mathrm{d}r \int_0^{2\pi}\mathrm{d}\theta = \left(\frac{r^2}{2}\right)\Big|_0^1 \theta\Big|_0^{2\pi} = \pi$$

$$p(r,\theta)\,\mathrm{d}r\,\mathrm{d}\theta = \frac{1}{\pi} r\,\mathrm{d}r\,\mathrm{d}\theta \rightarrow p(r,\theta) = \frac{r}{\pi}$$

so that we integrate to 1 instead of area

$$p(r,\theta) = p(r)p(\theta)$$ ⟵ r, θ **independent**



$$p(\theta) = \frac{1}{2\pi}$$

$$P(\theta) = \frac{1}{2\pi}\theta \qquad \theta = 2\pi\xi_1$$

$$p(r) = 2r$$

$$P(r) = r^2 \qquad r = \sqrt{\xi_2}$$

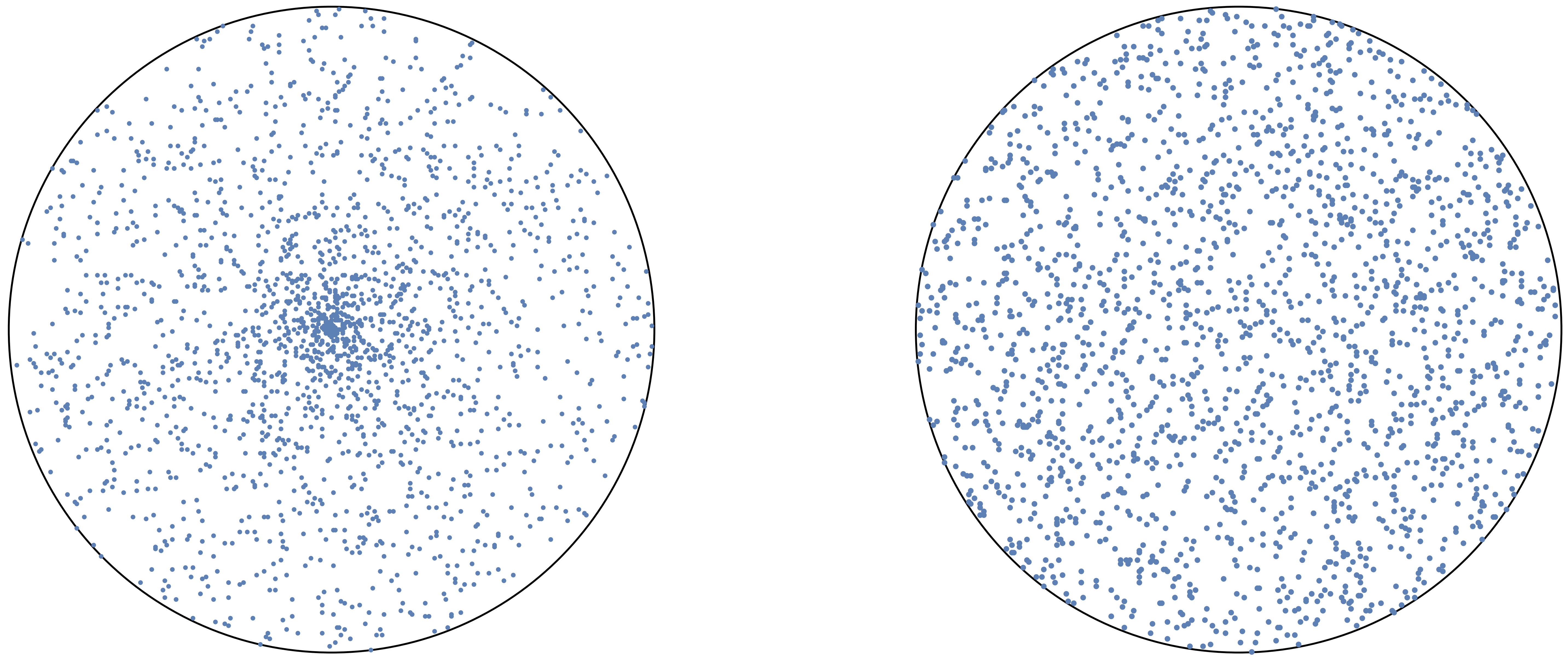
Uniform area sampling of a circle

WRONG

probability is uniform; samples are not!

RIGHT

probability is nonuniform; samples are uniform



WRONG:

$$\theta = 2\pi\xi_1$$

$$r = \xi_2$$

RIGHT:

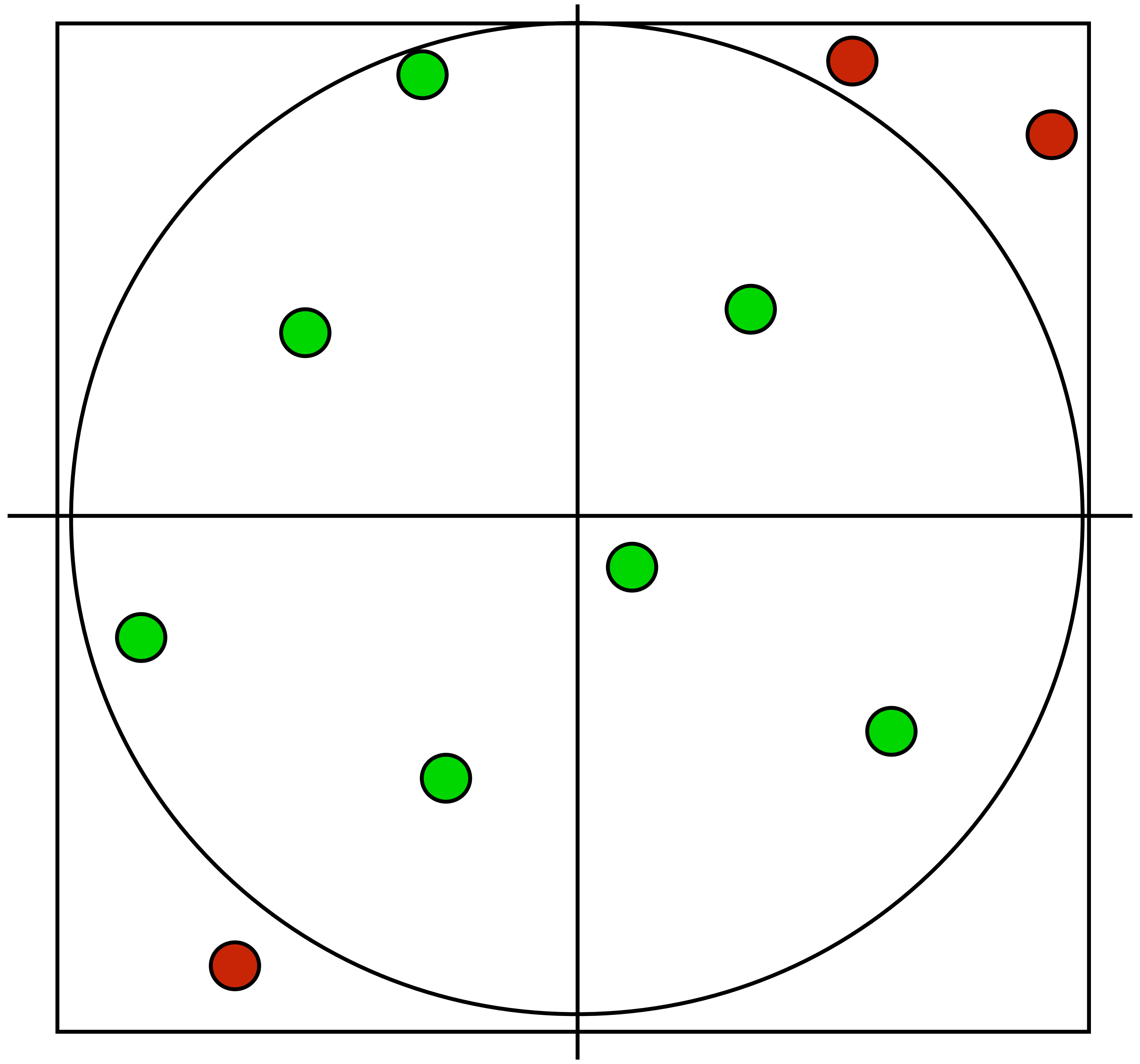
$$\theta = 2\pi\xi_1$$

$$r = \sqrt{\xi_2}$$

Uniform sampling via rejection sampling

Completely different idea: pick uniform samples in square (easy)

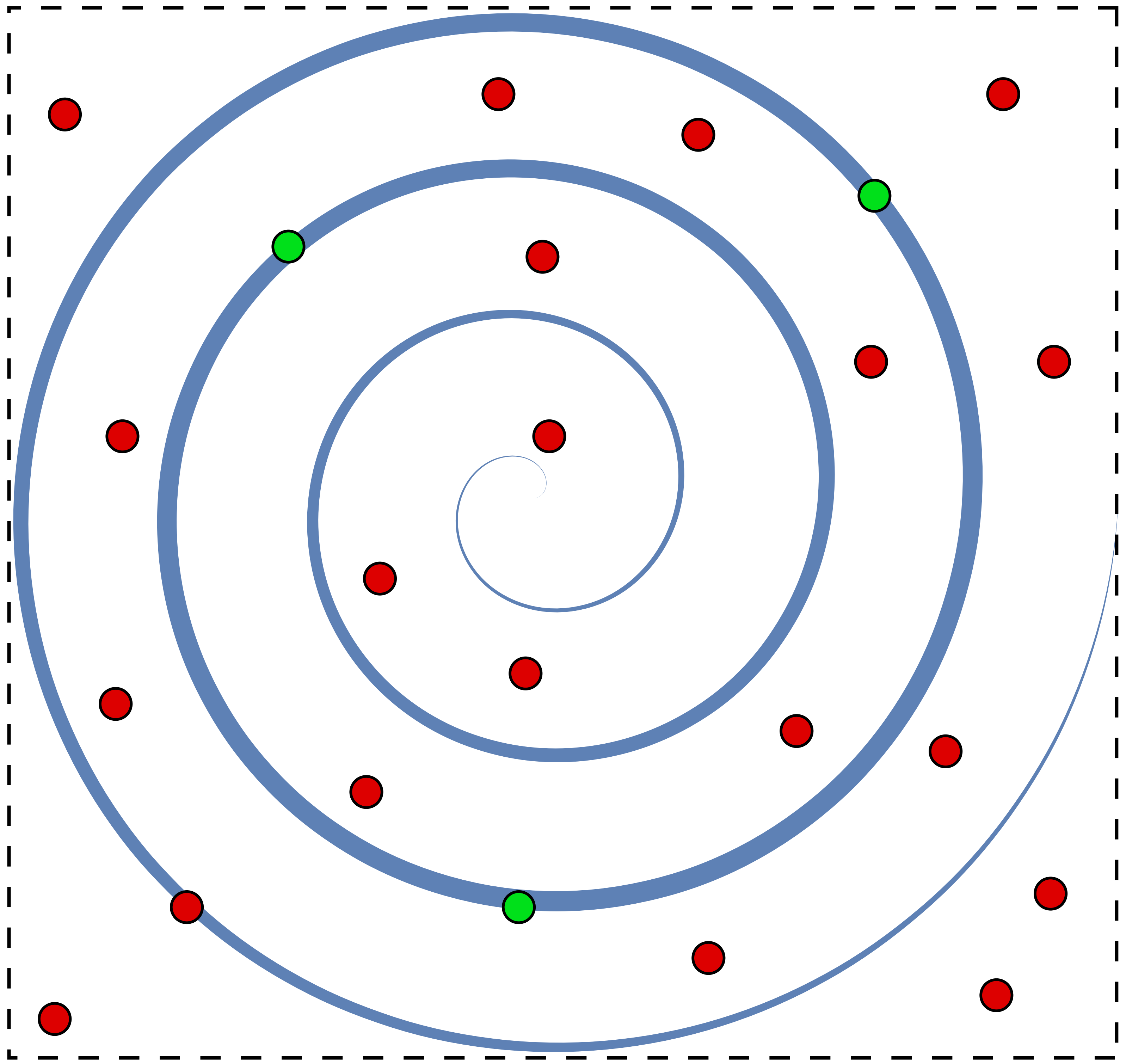
Then toss out any samples not in square (easy)



Efficiency of technique: area of circle / area of square

Efficiency of Rejection Sampling

- **If the region we care about covers only a very small fraction of the region we're sampling, rejection is probably a bad idea:**



Smarter in this case to "warp" our random variables to follow the spiral.

Expected Value

- What value does a random variable Y take, on average?
- E.g., consider a fair coin where heads = 1, tails = 0
- Equal probability of heads & is tails (1/2 for both)
- Expected value is then (1/2)*1 + (1/2)*0 = 1/2

Properties of expectation:

$$E\left[\sum_i Y_i\right] = \sum_i E[Y_i]$$

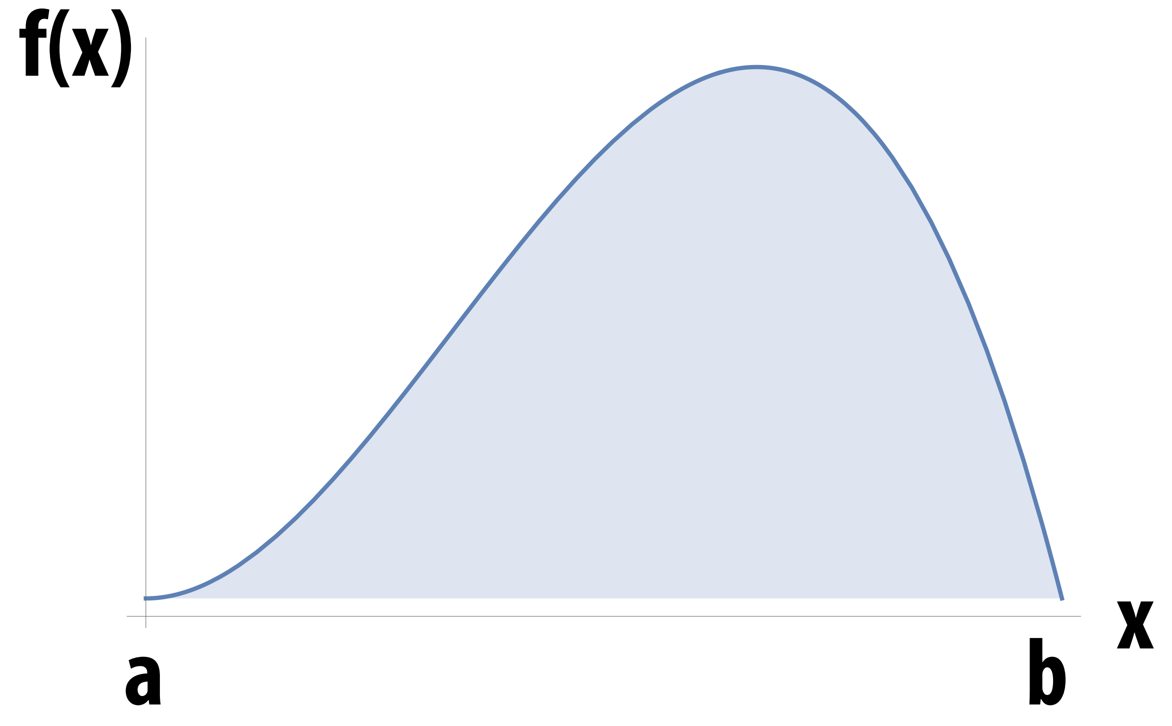
$$E[aY] = aE[Y]$$

(Can you show these are true?)

Putting it all together: Monte Carlo Integration

- **Definite integral**
  What we seek to estimate

$$\int_a^b f(x)dx$$



- **Random variables**
  X_i is the value of a random sample drawn from the distribution $p(x)$
  Y_i is also a random variable.

$$X_i \sim p(x)$$
$$Y_i = f(X_i)$$

- **Expectation of *f***
  For a *continuous* random variable

$$E[Y_i] = E[f(X_i)] = \int_a^b f(x)\, p(x)\, \mathrm{d}x$$

- **Estimator**
  Monte Carlo estimate of $\int_a^b f(x)dx$

$$F_N = \frac{b-a}{N} \sum_{i=1}^{N} Y_i$$

Assuming samples X_i drawn from uniform pdf. I will provide estimator for arbitrary PDFs next lecture.

Basic unbiased Monte Carlo estimator

Let's compute expected value of our numerical estimate of an integral:

$$
\begin{aligned}
E[F_N] &= E\left[\frac{b-a}{N}\sum_{i=1}^{N} Y_i\right] \\
&= \frac{b-a}{N}\sum_{i=1}^{N} E[Y_i] = \frac{b-a}{N}\sum_{i=1}^{N} E[f(X_i)] \\
&= \frac{b-a}{N}\sum_{i=1}^{N}\int_a^b f(x)\,p(x)\mathrm{d}x \\
&= \frac{1}{N}\sum_{i=1}^{N}\int_a^b f(x)\,\mathrm{d}x \\
&= \int_a^b f(x)\,\mathrm{d}x
\end{aligned}
$$

Assume uniform probability density for now

$$X_i \sim U(a,b)$$

$$p(x) = \frac{1}{b-a}$$

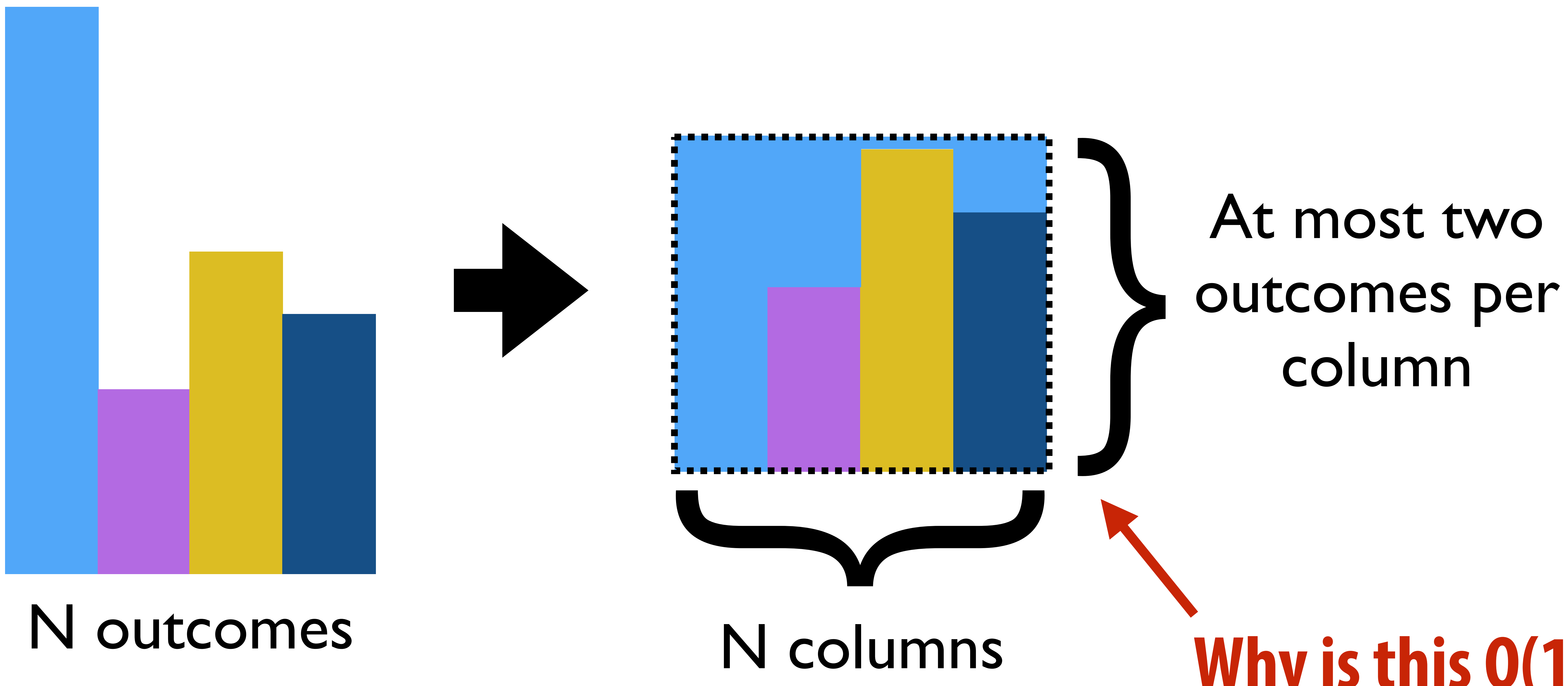
Since expected value is equal to the exact value of the integral we wanted, say this estimator is "unbiased"

Next Time: Monte Carlo Ray Tracing

Fun Stuff: The Alias Method

You can generate samples from *any* discrete probability distribution in O(1) time!

How? Pre-process to arrange the probability mass to perfectly cover a rectangle:



Why is this O(1) to sample?

Hey, is this pre-processing even *possible* in general?

(I did only show you one example…)