Numerical Integration

Phy 380

Dave Pawlowski

We've already covered numerical differentiation, but what about integration?

Various techniques:

Midpoint Rule

Trapezoidal Rule

Simpson's Rule

Monte Carlo techniques

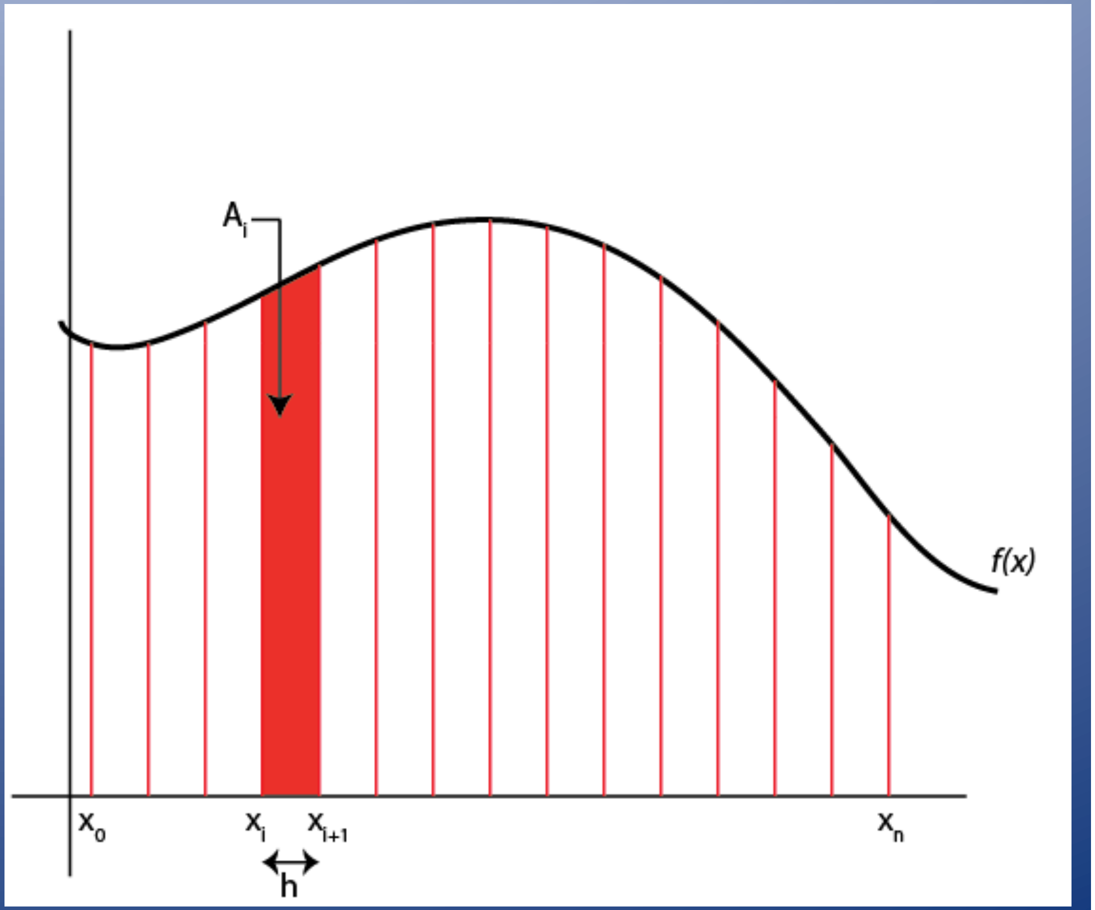
$$I = \int_a^b f(x)dx$$

How do you obtain the integral of a function that can't be obtained analytically?

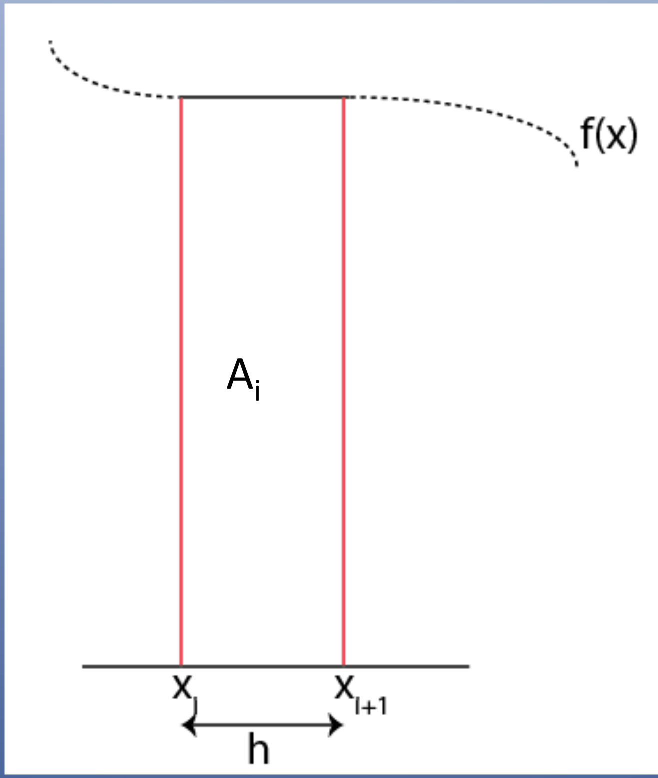
Graphical definition of the integral

- Given some function, the integral is simply the area under the curve.

Numerically, this means that we have to break up the area into discrete chucks, and sum the area of each chunk.



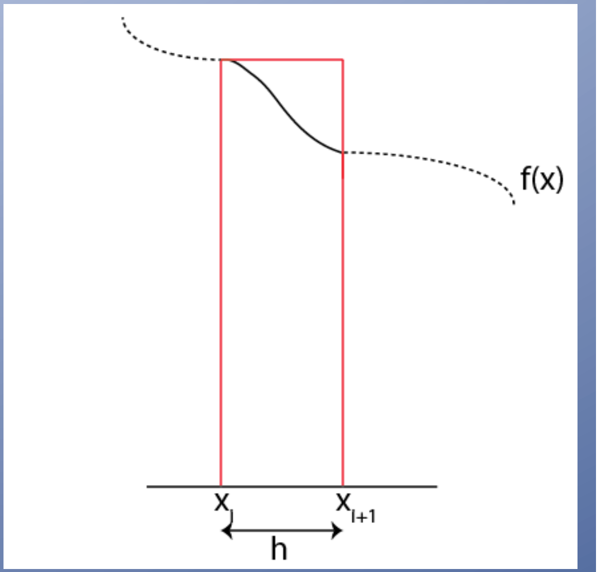
This is easy if the slope of the function evaluated at x_i and x_{i+1} is zero:



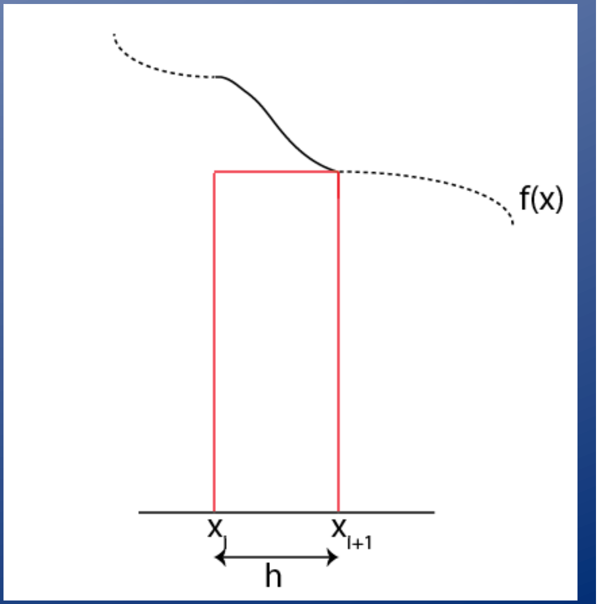
In this case, $A_i = h \times f(x_i)$

But if the function doesn't have zero slope, then which points should you use?

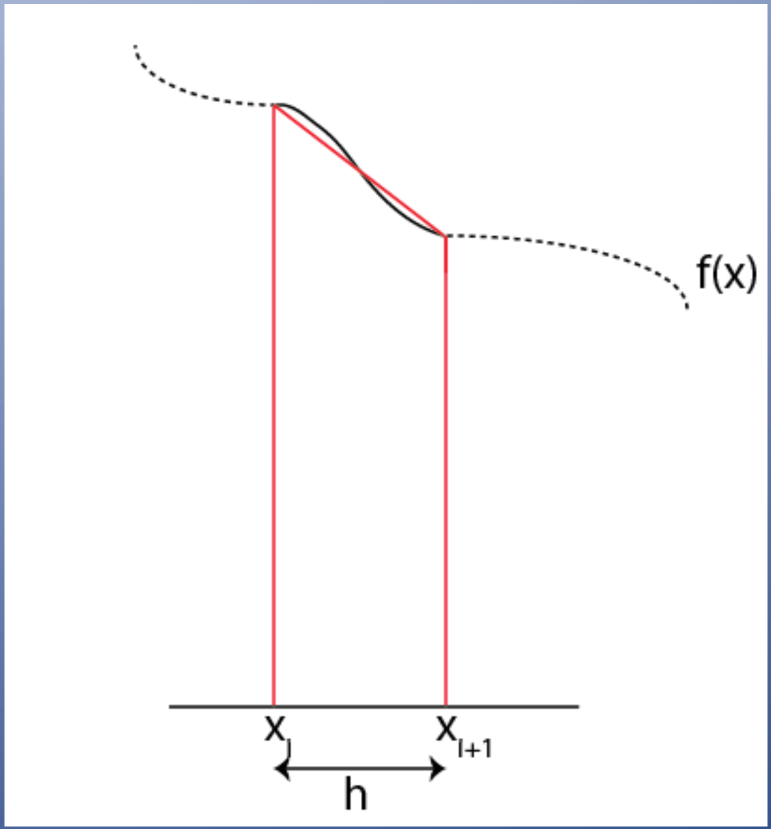
In this case, choosing $f(x_i)$ results in a rectangle that will over estimate the area under the curve.



And choosing $f(x_{i+1})$ results in a rectangle that will underestimate the area under the curve.



The trapezoid rule: Instead of using rectangles to calculate the areas of our discrete boxes, we can use trapezoids.



The area of this box can be can calculated nearly as easily as a rectangle.

$$I = \sum_{i=0}^{n-1} \frac{f(x_i) + f(x_{i+1})}{2} h$$

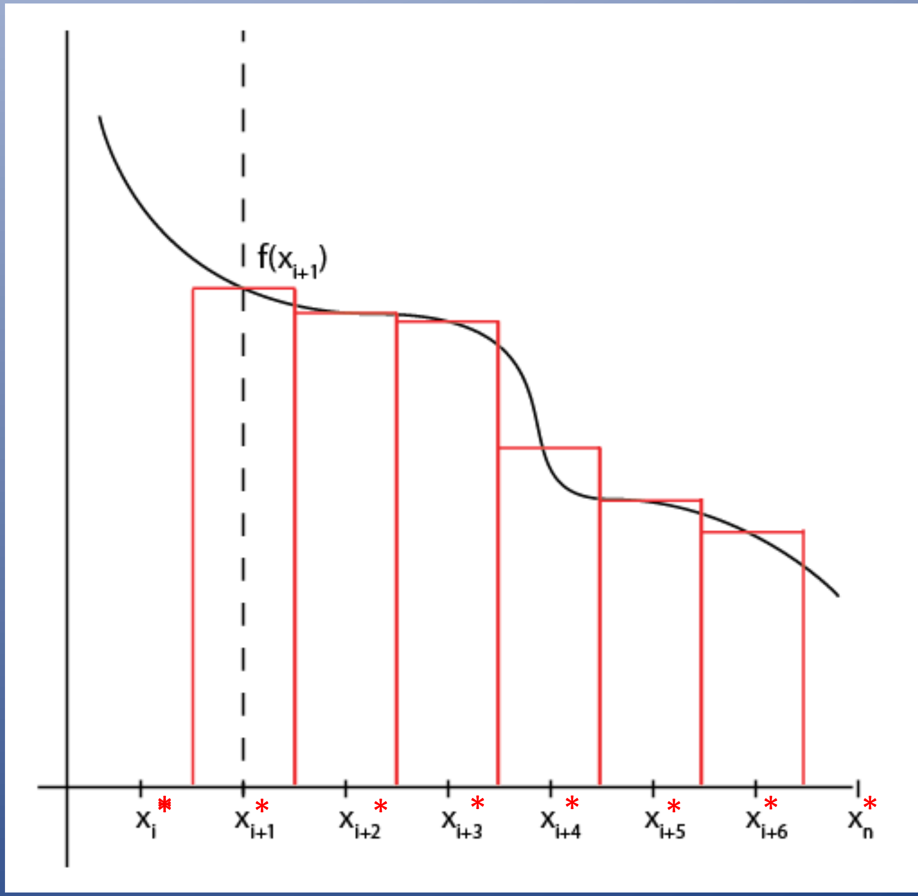
$$= \sum_{i=0}^{n-1} (\frac{f(x_i)}{2} + \frac{f(x_{i+1})}{2}) h$$

$$= (\frac{f(x_0)}{2} + f(x_1) + f(x_2) + \ldots + \frac{f(x_n)}{2}) h$$

Alternatively, we can treat the rectangles as if they are centered on the points of interest.

This is referred to as the midpoint rule, since it uses the midpoints of each rectangle to calculate the area under the curve. All that is left is to add up the individual areas:

$$I = \sum_{i=0}^{n-1} f(x_i^*)h$$



The problem is, we may not know the value of the function at the midpoint!

So far, the max errors of the two methods are

$$error \approx -\frac{h^3}{24} f''$$

midpoint

$$error \approx \frac{h^3}{12} f''$$

trapezoidal

The negative sign means that the when the function is concave up, then the approximation underestimates the solution.

By taking a weighted average of the two methods, we can effectively cancel out these errors, since they have opposite sign.

Try: $$I_s = \frac{2I_m + I_t}{3}$$

Let's use this to define the Simpson rule!

The Simpson Rule

$$I_2 \approx \frac{h}{3}\left(f(x_0) + 2\sum_{i=1}^{n/2-1} f(x_{2i}) + 4\sum_{i=1}^{n/2} f(x_{2i-1}) + f(x_n)\right)$$

$$\approx \frac{h}{3}\left(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \ldots + 4f(x_{n-1}) + f(x_n)\right)$$

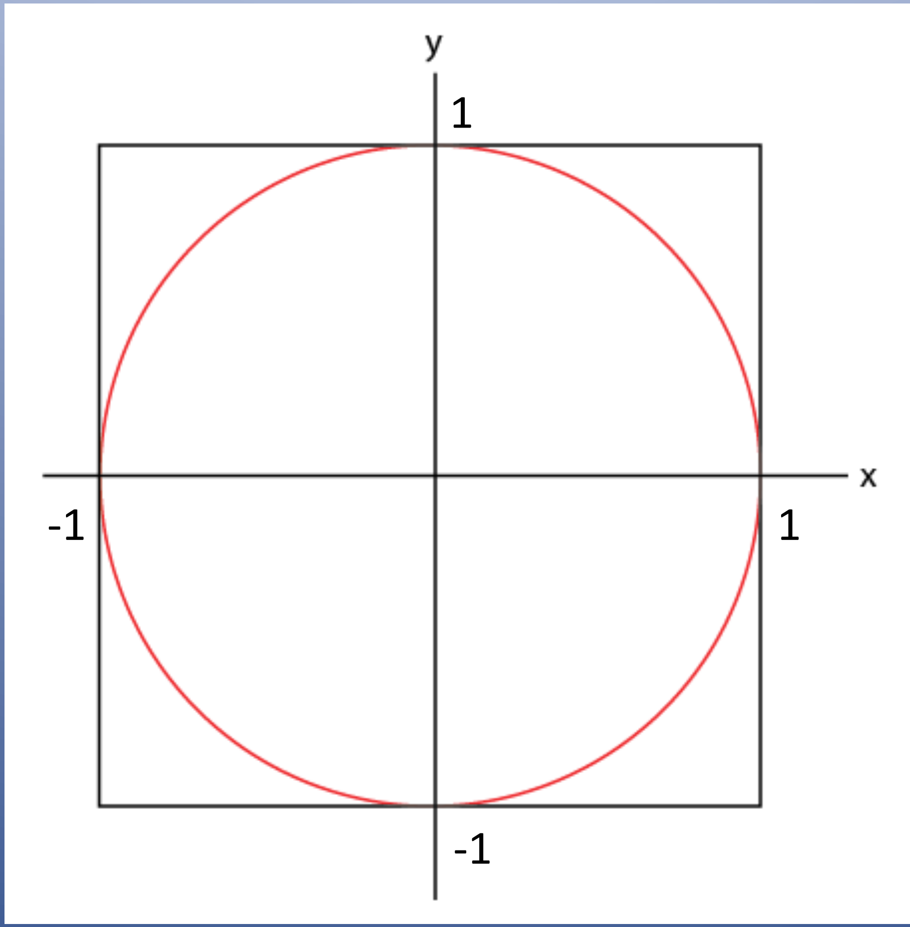
Note that n must be even.

The result has error ~h^5.

These techniques work very well for 1D systems. But as integrals are taken over more and more dimensions, it takes progressively more work to reach a good answer.

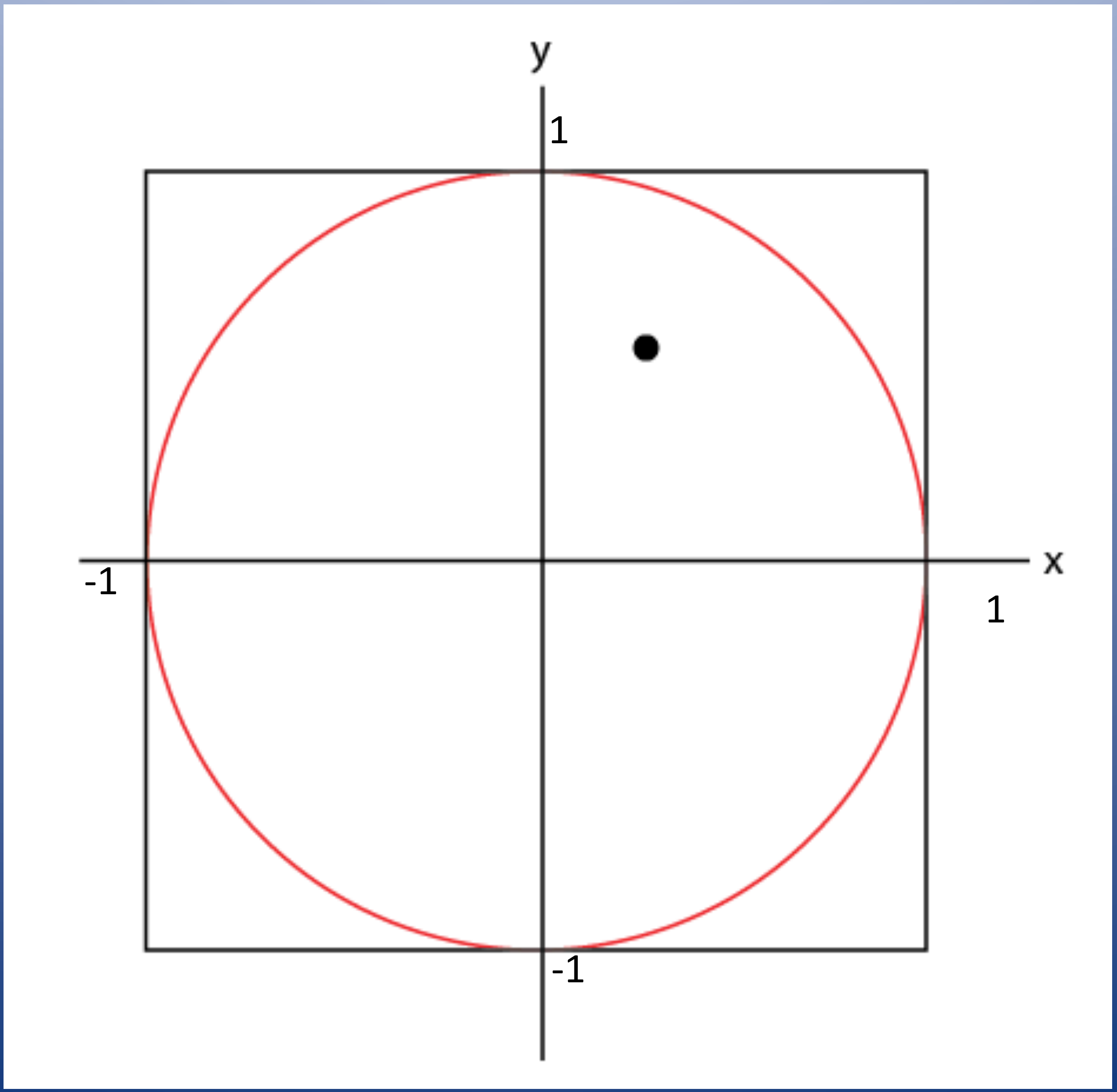
Alternative methods that perform nearly as well for integrating higher dimensional systems as they do for 1D systems are Monte Carlo methods- named after that one city on the Mediterranean.

Imagine a dartboard surrounded by a cardboard box…

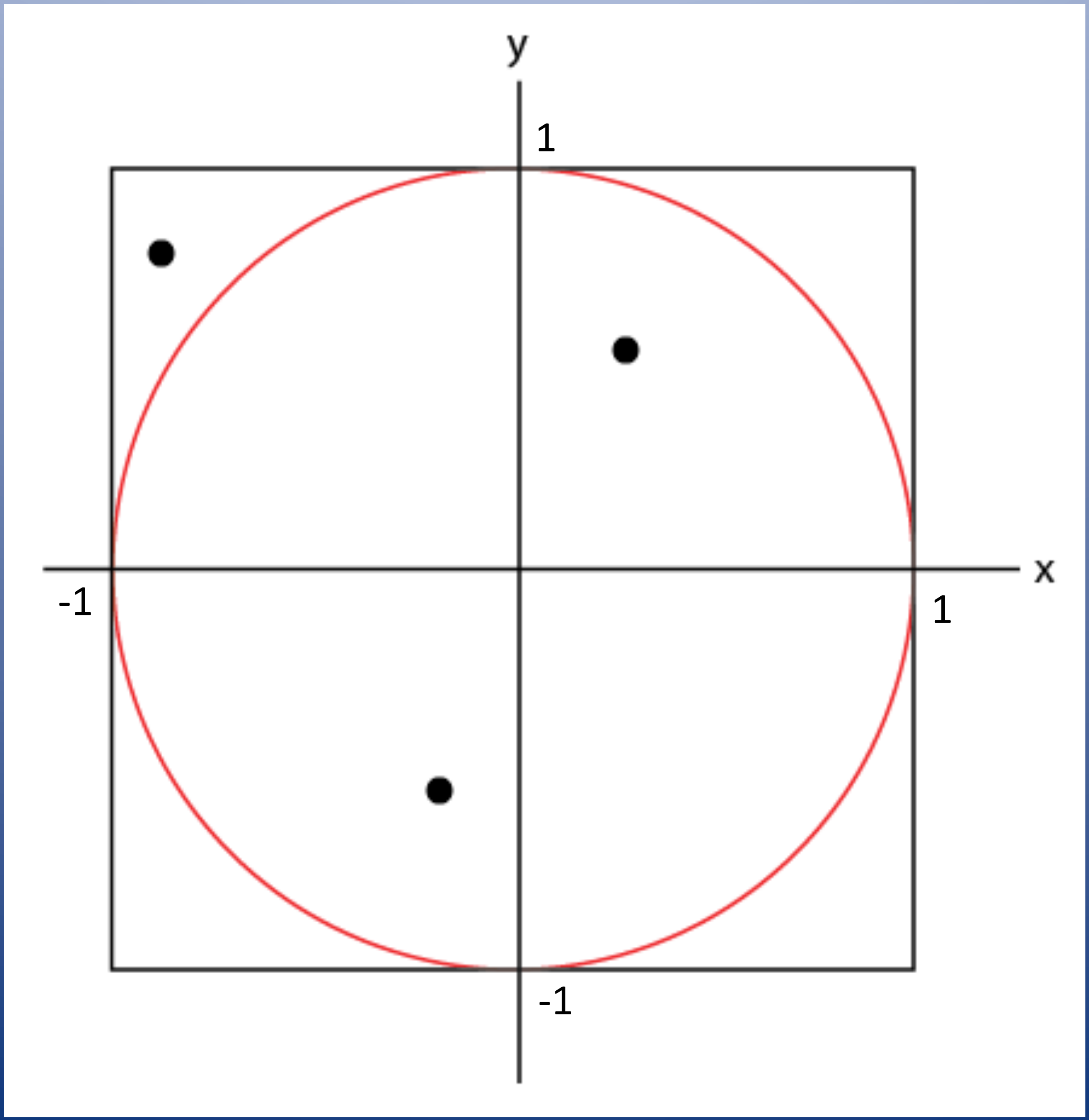


Can we estimate the area of the dartboard by throwing darts at it randomly?

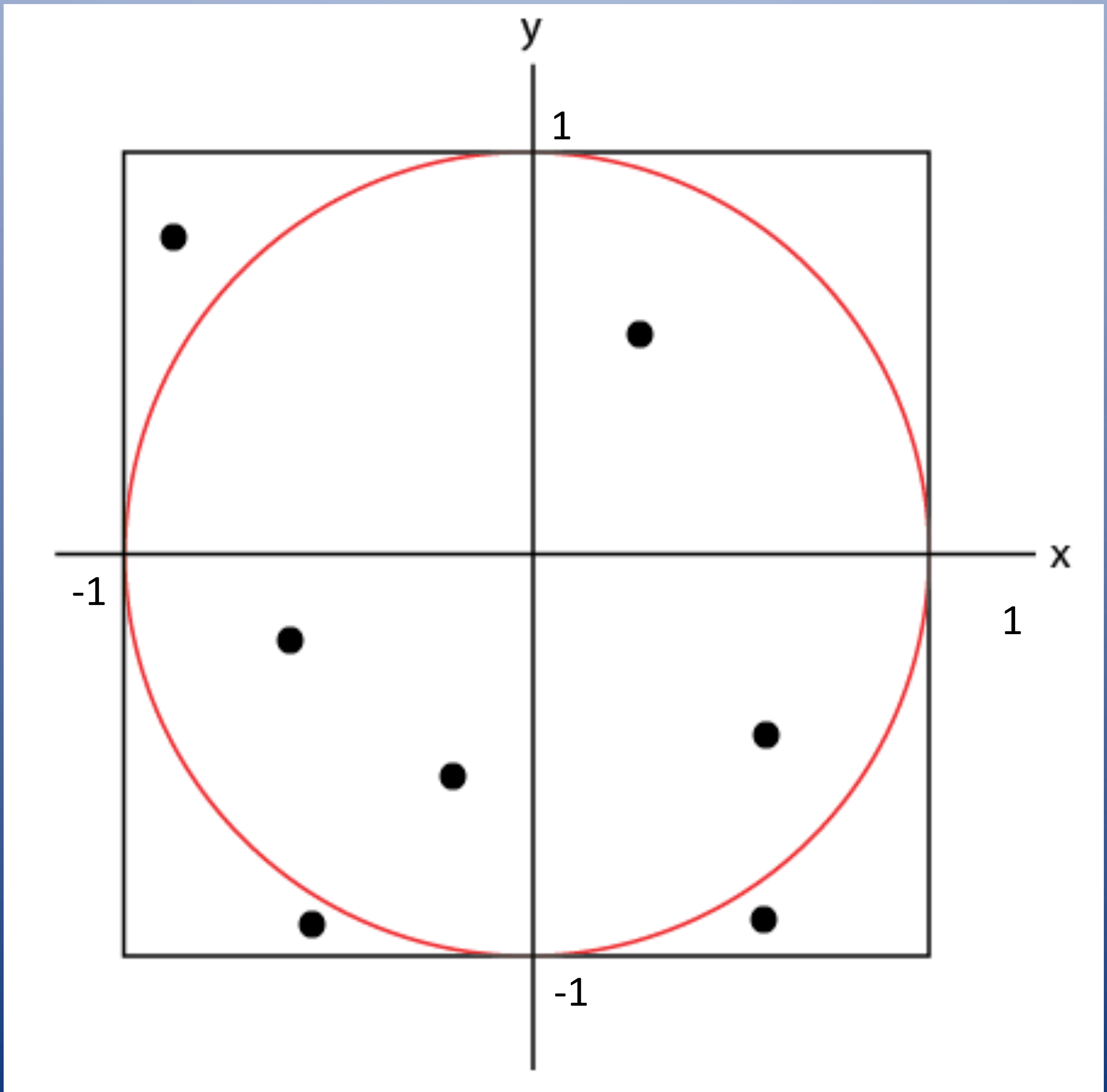
Inside the circle = 1, outside the circle = 0



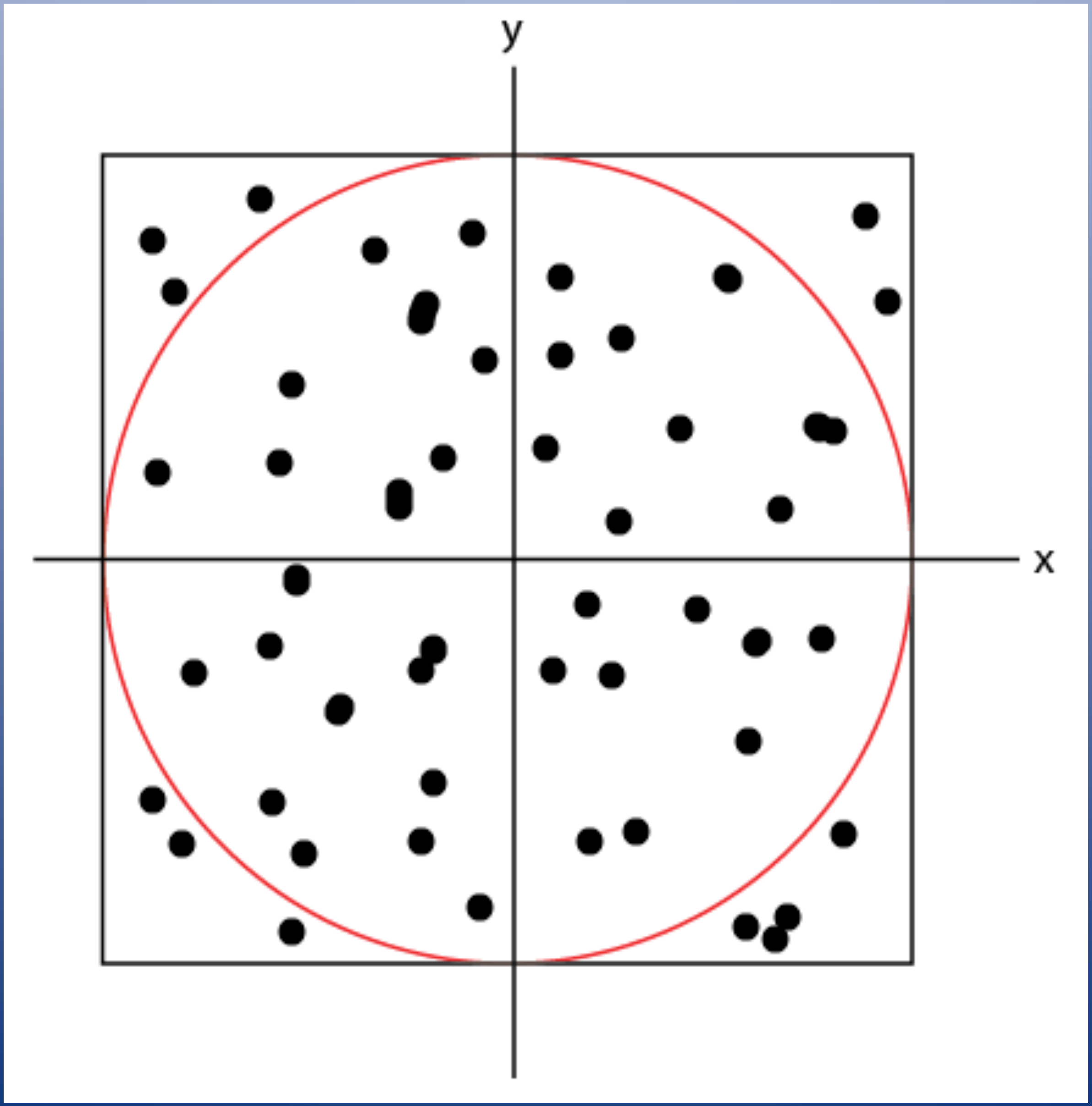
Inside the circle = 2, outside the circle = 1



Inside the circle = 4, outside the circle = 3

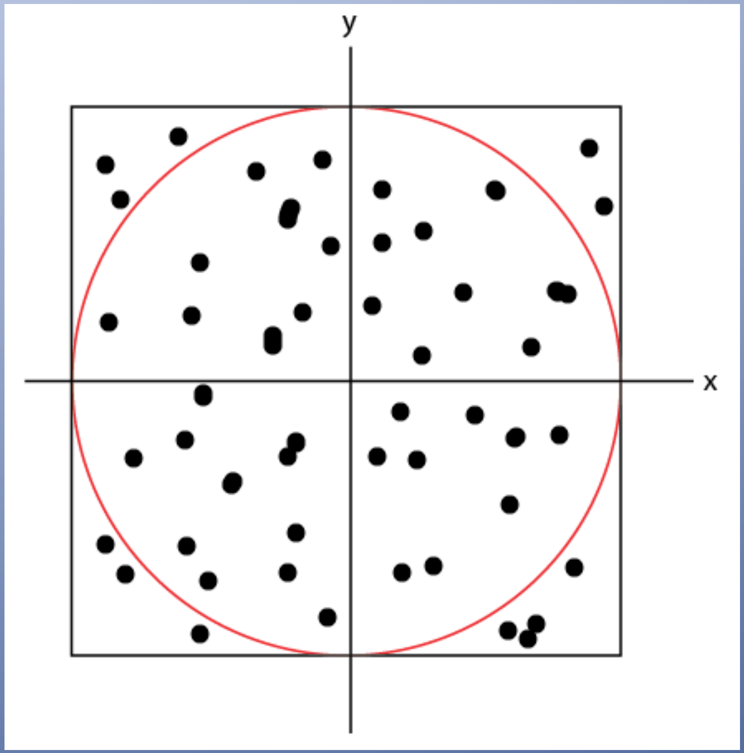


Inside the circle = 40, outside the circle = 10



So, 40 out of 50 darts it the circle…

That means that 80% of the darts that were in the square were also in the circle.



So, approximately: $A_c = .8 * A_s$

$$A_s = 4.0$$

$$A_c = 4.0 * 0.8 = 3.2 \approx \pi r^2 = 3.1415$$

In other words:

Pick a random coordinate (x,y,…) in the domain

Solve the function at that coordinate- f(x,y,…)

Is the y value that you picked less than f(x,y,…)?

Tally that as an “in”.

Repeat **n** times.

The integral can be approximated by:

the area (volume, etc) of the domain * in/n

The larger **n**, the better the approximation!

Monte Carlo techniques become more effective as the dimensionality of your problem goes up. For 1D and 2D integrals, it may not be computationally efficient. Beyond that MC methods may be a better/faster choice than other methods.