INTEGRATION

Ari Maller

NUMERICAL INTEGRATION

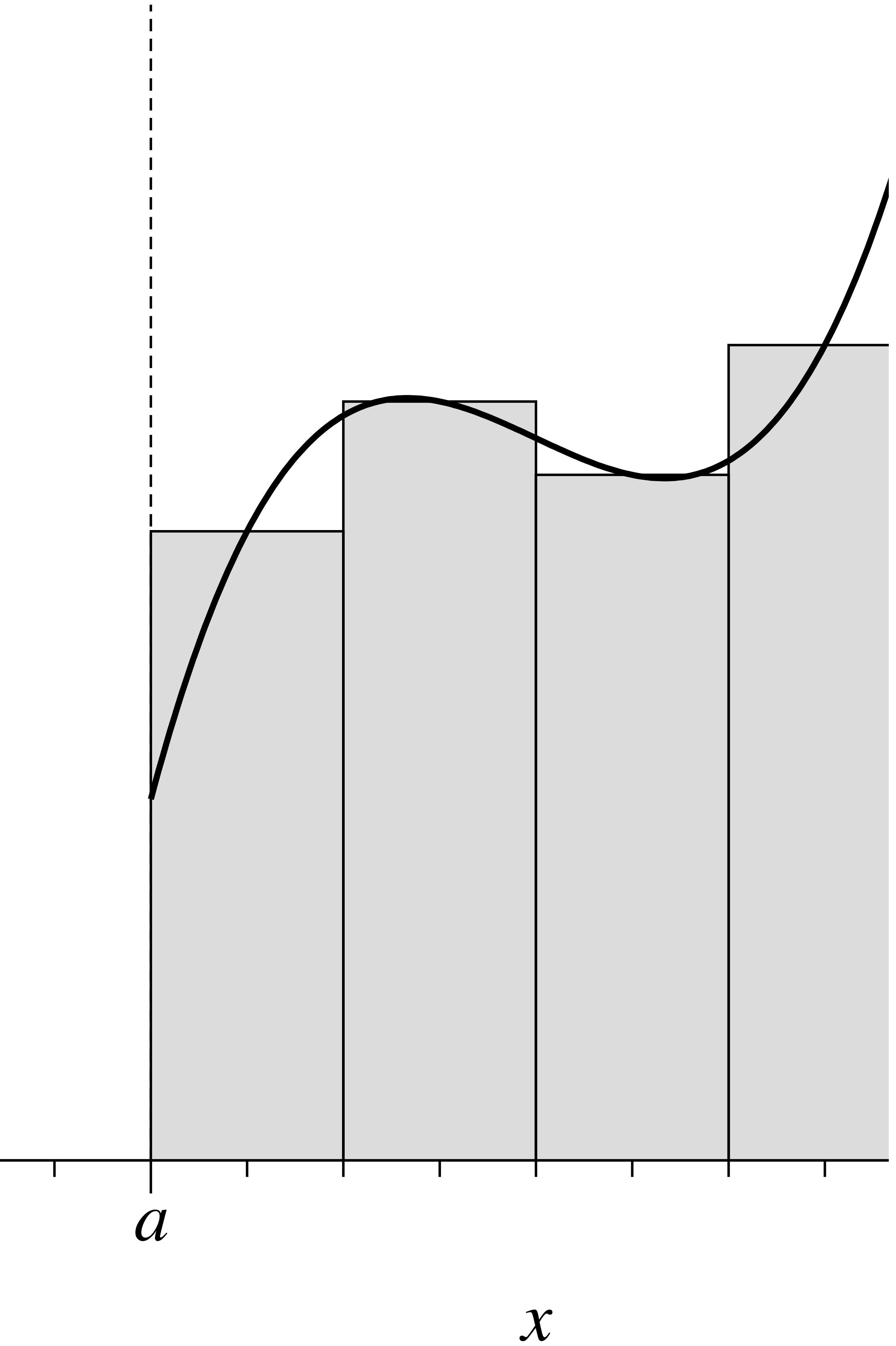
- Evaluating integrals is one of the most obvious things to do on a computer. Replacing the area under a curve with a large number of thin rectangles makes perfect sense on a computer (less sense when one is first learning calculus).
- The simplest situation is a one dimensional integral over a finite range. There are two common approaches to evaluating these types of integrals
  - Trapezoidal Rule
  - Simpsons Rule

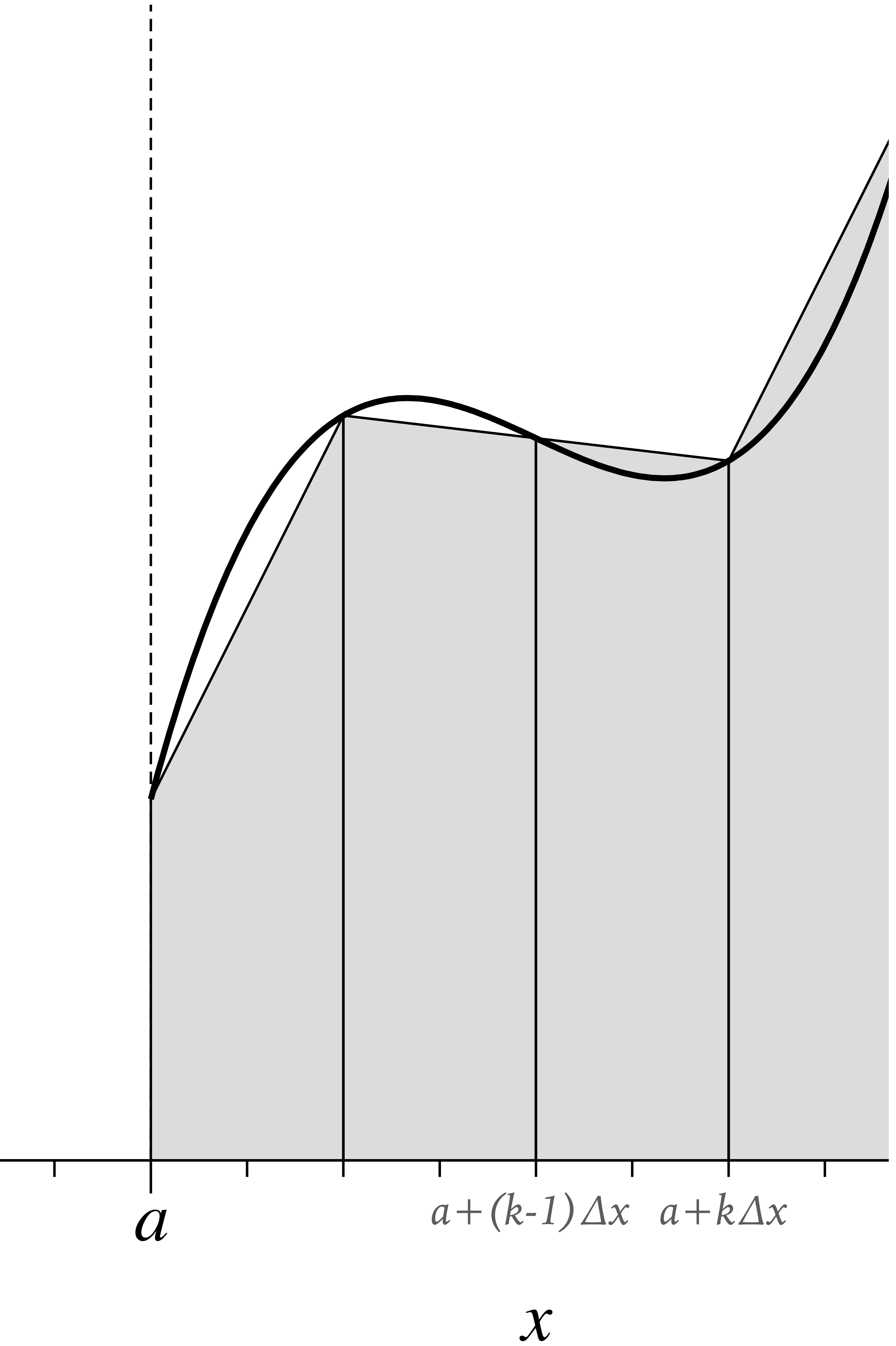
$$F = \int_a^b f(x)dx$$

RECTANGLE RULE

- The simplest way to approach an integral numerically would be to evaluate the function at N different points and then make a rectangle with the spacing between the points and then add up the rectangles.

$$F = \sum_{i=0}^{N} f(x_i)\Delta x$$





TRAPEZOIDAL METHOD

- Only slightly more complicated instead of adding rectangles we can add trapezoids.
- Since we are already evaluate f(x) at N points, this does not increase the number of evaluations.
- The area of each trapezoid is

$$A_k = \frac{1}{2}\Delta x[f(a+(k-1)\Delta x) + f(a+k\Delta x)]$$

- Adding up N trapezoids gives

$$F = \Delta x[\frac{1}{2}f(a) + \frac{1}{2}f(b) + \sum_{k=1}^{N} f(a+k\Delta x)]$$

EXAMPLE 5.1

- Let’s use the trapezoidal rule to calculate the integral of x^4-2x+1 from 0 to 2.
- Let’s start our program with 10 slices and then increase it to 100 and 1000.
- Solving the integral you should get 4.4.

$$F = \Delta x[\frac{1}{2}f(a) + \frac{1}{2}f(b) + \sum_{k=1}^{N} f(a + k\Delta x)]$$

EXAMPLE 5.1

```
def f(x):
    return x**4 - 2*x + 1
N = 10
a = 0.0
b = 2.0
h = (b-a)/N
s = 0.5*f(a) + 0.5*f(b)
for k in range(1,N):
    s += f(a+k*h)
print(h*s)
```

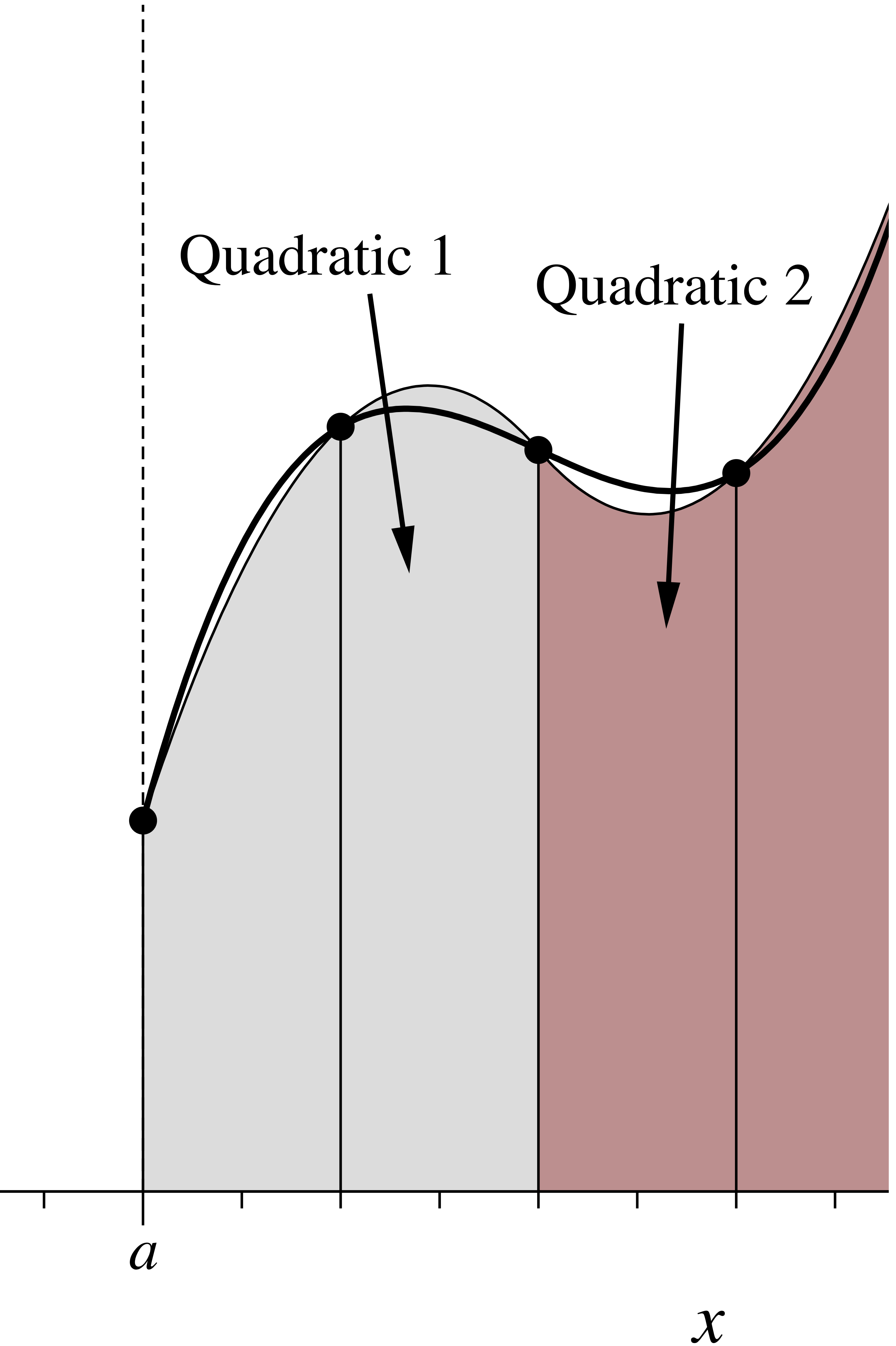
HIGHER ORDER FITS

- The trapezoidal method improves over the rectangle rule because x¹ fit to the curve is going to be better than a x⁰ fit.
- By this logic a quadratic fit will be even closer to the function we are trying to integrate then a linear fit.
- Fitting a quadratic to the curve is called Simpson's Rule.
- Let us consider three points given by -Δx, 0 and Δx. Then the value of our quadratic at these three points will be

$$f(-\Delta x) = A(\Delta x)^2 - B\Delta x + C \qquad f(0) = C \qquad f(\Delta x) = A(\Delta x)^2 + B\Delta x + C$$

- From these three equations we can no solve for A,B and C.

$$A = \frac{1}{(\Delta x)^2}[\frac{1}{2}f(-\Delta x) - f(0) + \frac{1}{2}f(\Delta x)] \quad B = \frac{1}{2\Delta x}[f(\Delta x) - f(-\Delta x)] \qquad C = f(0)$$



SIMPSON'S RULE

- The integral from -Δx to Δx of the quadratic is then

$$\int_{-\Delta x}^{\Delta x}(Ax^2+Bx+C)dx=\frac{2}{3}(\Delta x)^3+2C\Delta x$$
$$=\frac{1}{3}\Delta x[f(-\Delta x)+4f(0)+f(\Delta x)]$$

- So we have an algebraic expression for the area in this range. We can then sum up these areas from a to b. Note that we must choose the number of steps, N to be even in this case.

$$F(a,b)\simeq\frac{\Delta x}{3}\Big[f(a)+f(b)$$
$$+4\sum_{k=1}^{N/2}f(a+(2k-1)\Delta x)$$
$$+2\sum_{k=1}^{N/2-1}f(a+2k\Delta x)\Big]$$

EXERCISE 5.2

- Solve the same integral from before x^4-2x+1 from 0 to 2 but now using Simpson's rule with 10 slices.
- Now use 100 and 1000. How do your results compare to what we found using the trapezoidal rule?

$$F(a,b) \simeq \frac{\Delta x}{3}\Big[f(a) + f(b) + 4\sum_{k=1}^{N/2} f(a+(2k-1)\Delta x) + 2\sum_{k=1}^{N/2-1} f(a+2k\Delta x)\Big]$$

ERRORS ON INTEGRATION

- While rounding error exists in numerical integration it is not usually the main source of error.
- **Approximation error** is the dominant source of error. This is the error introduced because linear or quadratic fit only approximates the true function.
- Let us consider a Taylor expansion around a point x_{k-1} which is a distance h away from x. This gives

$$f(x) = f(x_{k-1}) + hf'(x_{k-1}) + \frac{1}{2}h^2 f''(x_{k-1}) + ...$$

- if we now integrate from x_{k-1} to x we get

$$\int_{x_{k-1}}^{x_k} f(x)dx = hf(x_{k-1}) + \frac{1}{2}h^2 f'(x_{k-1}) + \frac{1}{6}h^3 f''(x_{k-1}) + O(h^4)$$

ERRORS ON INTEGRATION

- if we instead expand around x$_k$ we would get

$$\int_{x_{k-1}}^{x_k} f(x)dx = hf(x_k) - \frac{1}{2}h^2 f'(x_k) + \frac{1}{6}h^3 f''(x_k) - O(h^4)$$

- now averaging the two expressions gives

$$\int_{x_{k-1}}^{x_k} f(x)dx = \frac{1}{2}h(f(x_{k-1}) + f(x_k)) - \frac{1}{4}h^2(f'(x_{k_1}) - f'(x_k)) + \frac{1}{12}h^3(f''(x_{k-1}) + f''(x_k)) - O(h^4)$$

- we can now sum over k for all slices from a to b

$$\int_a^b f(x)dx = \frac{1}{2}h\sum_{k=1}^{N}[f(x_{k-1}) + f(x_k)] + \frac{1}{4}h^2[f'(a) - f'(b)]$$

$$+\frac{1}{12}h^3\sum_{k=1}^{N}(f''(x_{k-1}) + f''(x_k)) + O(h^4)$$

- The second term is the trapezoidal method that we have used to numerically estimate the integral, which means the remaining terms are the difference between that estimate and the correct value.

- The sum in the second term disappears because you are adding and subtracting each f'(x$_k$).

EULER-MACLAURIN FORMULA FOR THE ERROR

- The third term is just the trapezoidal approximation for f''(x)

$$\frac{1}{12}h^3\sum_{k=1}^{N}[f''(x_{k-1})+f''(x_k)] = \frac{1}{6}h^2\int_a^b f''(x)dx + O(h^4) = \frac{1}{6}h^2[f'(b)-f'(a)] + O(h^4)$$

- substituting this back into our result we get

$$\int_a^b f(x)dx = \frac{1}{2}h\sum_{k=1}^{N}[f(x_{k-1})+f(x_k)] + \frac{1}{12}h^2[f'(a)-f'(b)] + O(h^4)$$

- Thus the error, the difference between the integral and the trapezoidal method, first term or terms of order less than h^4 is

$$\epsilon = \frac{1}{12}h^2|f'(a)-f'(b)|$$

ORDER OF ACCURACY

- The equation for the error tells us that the trapezoidal rule is a *first-order* integration scheme. It is accurate to terms involving h and the first term of its error is h^2.
- If we do similar analysis for the Simpson's rule we would find the first term in the error is give by

$$\epsilon = \frac{1}{90} h^4 [f'''(a) - f'''(b)]$$

- The Simpson's method is a *third-order* integration method. It is accurate to terms involving h^3 and the first term of the error is h^4.
- Note that because the error for the trapezoidal rule depends on the first derivative while for Simpson's rule it depends on the third derivative in some cases the error for Simpson's integration can be much larger than trapezoidal.

PRACTICAL ESTIMATION OF ERRORS

- Our estimates of the error in our integration schemes depend on knowing the derivatives at a and b. However, often we will want to integrate values which may not be a mathematical function.
- In this case we can estimate the error by evaluating the integral with a certain number of steps N_1 and step size h_1. And then evaluating the integral again with say double the number of steps $N_2 = 2N_1$ and half the step size $h_2 = 1/2\ h_1$.
- If the true value of the integral is I, and the difference between the true value and the numerical value is ch^2, we will have

$$I_1 + ch_1^2 = I_2 + ch_2^2 \quad => I_2 - I_1 = ch_1^2 - ch_2^2 = 3ch_2^2$$

- the error on the second integration would be $\varepsilon_2 = 1/3|I_2 - I_1|$. Similarly for Simpson's method we would get $\varepsilon_2 = 1/15|I_2 - I_1|$.

EXERCISE 5.6

- Modify your program that calculates the integral of x^4-2x+1 so that it can estimate the error with 20 steps.
- To do this you'll have to run it with 10 steps and then 20 steps.

NUMBER OF STEPS

- How do we decide the number of steps to take in evaluating our integral.
- If we take h to be very small we can make the approximation error very small.
- But if h becomes too small then round off error begins to grow. We can even choose h so small that when we add the next step to our sum it doesn't actually increase it.
- Clearly decreases h is of no help once the approximation error is of order the rounding error. For the trapezoidal rule

$$\frac{1}{12}h^2[f'(a) - f'(b)] \simeq C \int_a^b f(x)dx$$

- In Python where C~10^{-16} this gives N~10^{8}. For Simpson's rule we get N~10^{4}.

ADAPTIVE INTEGRATION

- Going to machine precision can take a long time if you are evaluating many integrals and is rarely the accuracy you actually need.
- Another choice is estimate the accuracy by running two values of the number of steps and then continue increasing N until you get the desired accuracy.
- If we double the number of steps then we don't actually need to evaluate each of those steps because half of them were in the previous calculation.
- This is called nested steps and makes adaptive integration only take about as long as the final number of steps you end up using.

ADAPTIVE INTEGRATION

- Because adaptive nested steps has almost the same computational cost as a fixed number of steps this is almost always the way integrals are evaluated.
- A starting value is chose for the number of steps N and the function is evaluated at those N points.
- Then you use 2N steps, but you only have to evaluate the function for N new points. The number of points keeps being doubled and each time the error can be estimated so the process stops when the error is less than a specified value.
- This insures you never use too many steps, more steps than needed to get the error you are looking for. This helps tremendously with speed, especially if you are performing many integrals.

ROMBERG INTEGRATION

- One can imagine doing even better than adaptive stepping by using error estimates to improve our calculation of the integral.
- Remember for the trapezoidal rule we had for the error

$$ch_i^2 = I_i - I_{i-1}$$

- but since the integral $I = I_i + ch_i^2 + O(h^4)$ we can write the true value as

$$I = I_i + \frac{1}{3}(I_i - I_{i-1}) + O(h^4)$$

- This expression is now accurate to 3rd order, as accurate as Simpson's method.

ROMBERG INTEGRATION

- Let's refine out notation. Let the result of our integration be called R with a first subscript denoting the number of adaptive steps we've taken and a second subscript denoting the order of corrections we have done to calculated value. So

$$R_{i,1} = I_1 \qquad R_{i,2} = I_i + \frac{1}{3}(I_i - I_{i-1}) = R_{i,1} + \frac{1}{3}(R_{i,1} - R_{i-1,1})$$

- which tells us the true value of the integral should be

$$I = R_{i,2} + c_2 h_i^4 + O(h_i^6)$$

- where c_2 is a new constant and there is no h^5 term because only even powers contribute. We can also equate to

$$I = R_{i-1,2} + c_2 h_{i-1}^4 + O(h_{i-1}^6) = R_{i,2} + 16 c_2 h_i^4 + O(h_i^6)$$

- and use this to eliminate the h^4 term.

ROMBERG INTEGRATION

$$c_2 h_i^4 = \frac{1}{15}(R_{i,2} - R_{i-1,2} + O(h_i^6)$$

$$I = R_{i,2} + \frac{1}{15}(R_{i,2} - R_{i-1,2}) + O(h_i^6)$$

- We now have an estimate of the integral that is accurate to fifth order in h!
- And there is no reason we have to stop here. For trapezoidal computation we make with a different step size we add an additional correction to our error.
- If we calculate I_1, I_2, I_3 and I_4 then we can use the difference between them to determine $R_{4,4}$, a correction to I_4 based on the values of I_1, I_2 and I_3.
- If we are using adaptive step sizes then we are calculating these anyway so Romberg integration requires very little extra computation.

RICHARDSON EXTRAPOLATION

- Romberg integration in essence is making a series expansion around the step size h.
- It will work will if this series converges quickly. If the series doesn’t for many terms then you don’t gain anything over simple adaptive trapezoidal rule.
- For well behaved integrals this should work well. But if the integrand is poorly behaved, contains wild fluctuations, singularities or is very noisy, Romberg integration is not a good method.
- Romberg integration is an example of *Richardson extrapolation* where higher order estimates of quantities are calculated iteratively. We will return to this method when discussing solutions to differential equations.

HIGHER ORDER METHODS

- We have seen that the trapezoidal method fits the integrand with a line or first order polynomial.
- And Simpson's method uses a quadratic or 2nd order polynomial.
- There is nothing to stop us using higher order polynomials. A general description of this method can be written as

$$\int_a^b f(x)dx \simeq \sum_{k=1}^{N} w_k f(x_k)$$

- where w_k is the weight applied to each term in the sum. Higher order calculations end up just being a change in the weights, w_k. These methods as a whole are called Newton-Cotes formula where the integrand is evaluated at equally spaced points.

QUADRATURE

- So far we have looked at integration methods that use evenly spaced sampling of the function. These approaches can be referred to Newton - Cotes integration where the trapezoid rule and Simpsons rule are examples of Newton - Cotes schemes.
- Alternatively, one could think of not sampling the function evenly. In fact, one might suspect that one could get better results with fewer points if one chooses the points wisely. When the points are chosen in this way the approach is called Quadrature.

GAUSSIAN QUADRATURE

- So far we have just adjusted the weights given to each of our evaluation of the integrand, now we turn to adjusting where we evaluate the integrand.
- Suppose we are given a nonuniform set of N points x_k and we want to develop an integration rule that will only use the value of the function $f(x_k)$ at those points.
- This could be for example a set of experimental data where we don't have the luxury of simply increasing N to larger and larger values.
- If we fit a N-1 polynomial to our points then we will be able to integrate it exactly.

GAUSSIAN QUADRATURE

- Consider the following function called the interpolating polynomial

$$\phi_k(x) = \prod_{m=1}^{N} \frac{(x - x_m)}{x_k - x_m}$$

- This is a N-1 polynomial. The value of this function at x_m will be the Kronecker delta, that is 1 for k=m and 0 otherwise.

$$\phi_k(x_m) = \delta_{km}$$

- We can use this function to find a fits the integrand f(x) at the same points.

$$\Phi(x_m) = \sum_{k=1}^{N} f(x_k)\phi_k(x_m) = \sum_{k=1}^{N} f(x_k)\delta_{km} = f(x_m)$$

- so we can now evaluate our integral.

$$\int_a^b f(x)dx \simeq \int_a^b \Phi(x)dx = \int_a^b \sum_{k=1}^{N} f(x_k)\phi_k(x)dx = \sum_{k=1}^{N} f(x_k) \int_a^b \phi_k(x)dx$$

GAUSSIAN QUADRATURE

- So we find that the weights we need when evaluating the integral are

$$w_k = \int_a^b \phi_k(x)dx$$

- This might seem like not much of a victory, because there are no general closed form formula for the interpolating polynomial. So we've simply replaced our unknown integral with a different unknown integral.

- However, the difference is that we only have to calculate these weights over a specific range once, and then we can use them over and over to calculate different integrals.

- It is even better than this because the weights can be mapped between different intervals. Traditionally they are calculated for [-1,1], but we can just slide any interval [a,b] over to that range.

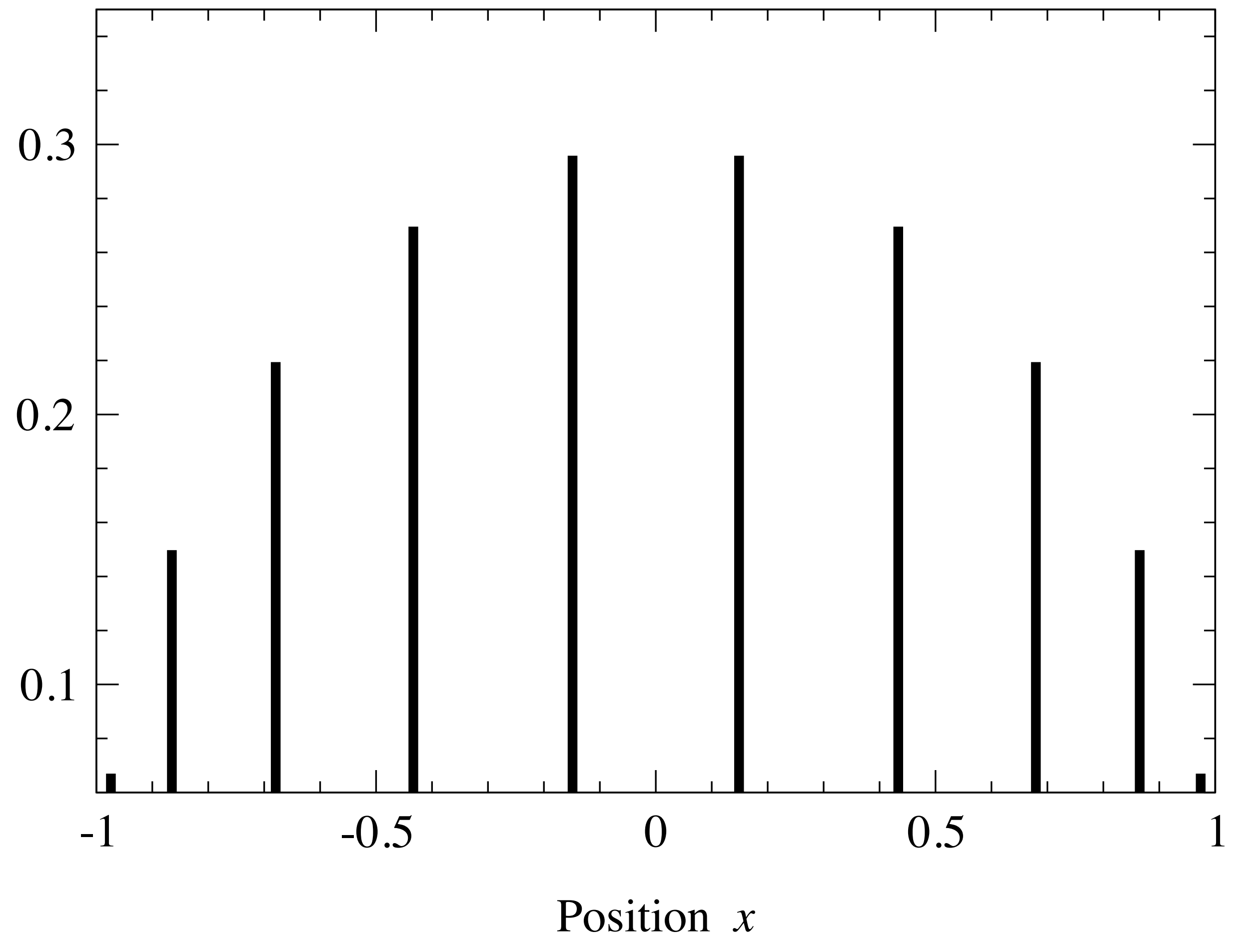
$$x'_k = \frac{1}{2}(b-a)x_k + \frac{1}{2}(b+a) \qquad w'_k = \frac{1}{2}(b-a)w_k$$

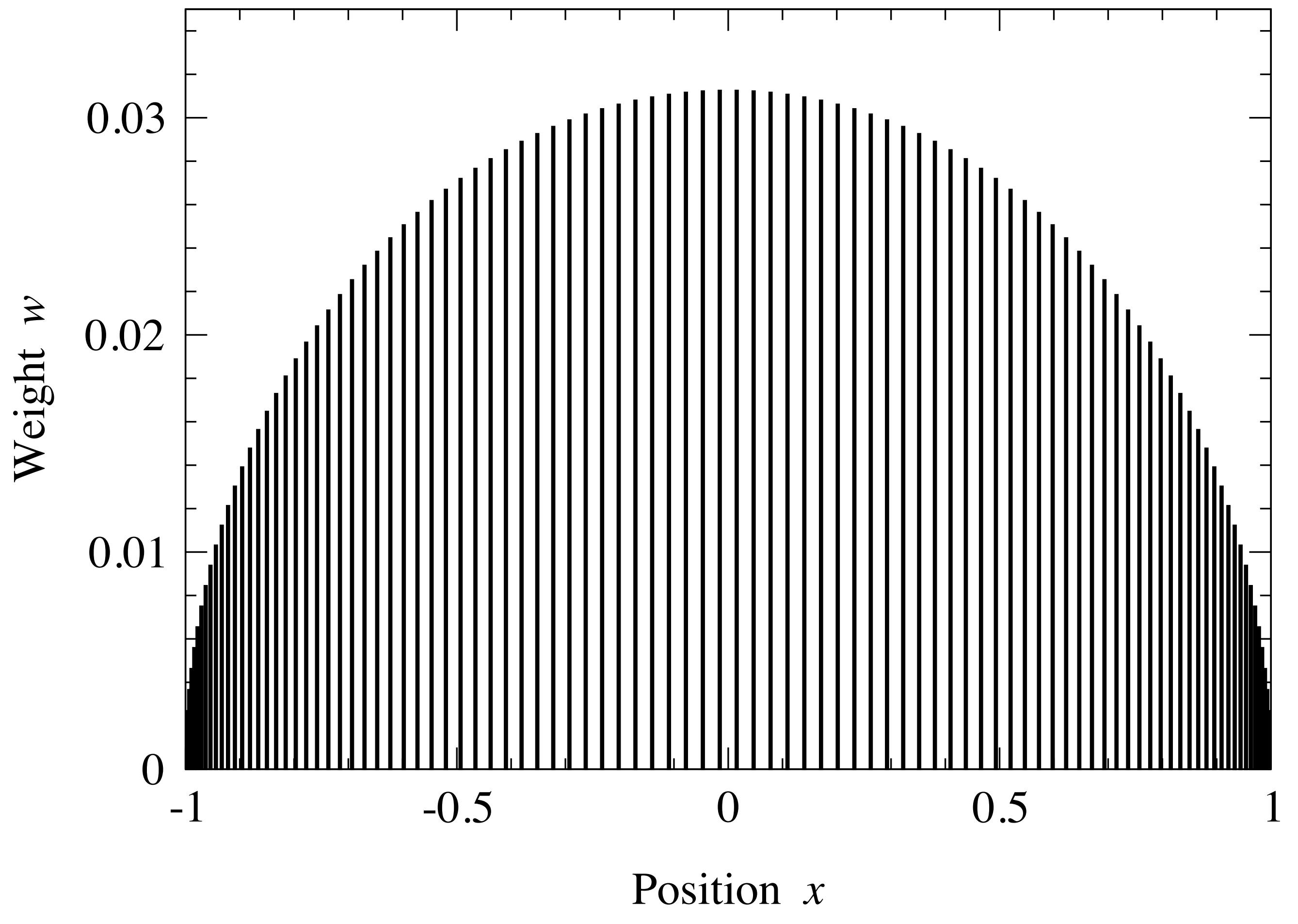
GAUSSIAN QUADRATURE

- We now have a method that will give us the weights given the points we are going to evaluate in our integrand, but how should we choose the points to evaluate.
- It turns out that the best choice is the zeros of the Nth Legendre polynomial $P_N(x)$, rescaled to the window of integration. Proof is given in Appendix C in the book.
- The weights to use are then

$$w_k = \left[\frac{2}{(1-x^2)}\left(\frac{dP_N}{dx}\right)^{-2}\right]_{x=x_k}$$

- This is called Gaussian quadrature, there are under quadratures using different basis functions that work in a similar way.
- While the math is much more complicated the code to run this integration isn't that bad and can give very accurate answers in a small number of steps.





The sample points and weights for Gaussian quadrature for N=10 and N=100 points. While it took some effort to get here, once these values are calculated you can use them now forever to evaluate integrands over this range with these number of points. Notice that the points are not uniformly spaced.

EXAMPLE 5.2

- We can use Gaussian quadrature to evaluate the integrand we have been using this whole chapter
- f(x) = x^4 - 2x + 1
- We will only need 3 steps this time. To get the points and weights however we will need a package gaussxw (http://www-personal.umich.edu/~mejn/computational-physics/).
- Once you get the points and weights you just sum the function at those points with those weights.

$$\int_a^b f(x)dx \simeq \sum_{k=1}^{N} w_k f(x_k)$$

ERROR ON GAUSSIAN QUADRATURE

- The formula for the error from Gaussian quadrature is ungainly and not easy to use in practice. Roughly the error improves by c/N^2 for each increase in N. That is going from 10 to 11 points gives a hundred fold increase in accuracy. Doubling the number of points gives a reduction in error of roughly N^{-2N}.
- If we make two determinations of an integral with Gaussian quadrature then the one with more points is so much more accurate we can treat it like it has zero error and estimate the error as the difference between our two evaluations. This is really the error of the evaluation with fewer points, but none the less is a way to estimate an error. One can then increase the number of points till you have the accuracy you desire.
- What this means is one can start with say $N=10$ points for a Gaussian quadrature and then keep doubling them till the error is below some limit. However, because the points are not nested each time one has to determine the points, weights and values of the integrand, so it is not nearly as effective as adaptive stepping for a Newton-Cotes formula method.

CHOOSING AN INTEGRATION METHOD

- We have discussed four methods for numerical integration; trapezoidal, Simpson's, Romberg and Gaussian quadrature. Now the question is which one should you use?
- The answer, as always, depends on exactly what you want to do. Higher order methods will give you an accurate answer in few steps if the integrand is well behaved.
- For problematic integrands higher order methods may not converge well and simpler methods like trapezoidal may before better.
- The key feature is really how many points are needed to get a good sense of what the integrand looks like. If you need 10,000 then Romberg and Gaussian will fail miserably with 10. If 10 basically gives you a good sketch of the function then they will do well.
- Whether you need even spacing or not will also depend on the situation.

INFINITE RANGE INTERVALS

- So far we have discussed integration between two points a and b, but what if we want to evaluate an integral between 0 and infinity.
- Then dividing the space by N will not help. Either each step will be infinitely large or if we choose a finite h then we will need an infinite number of steps to evaluate the integral.
- To deal with integrals over an infinite range we use a change of variables to make the range finite.

$$x = \frac{z}{z-1} \qquad \int_0^\infty f(x)dx = \int_0^1 \frac{1}{(z-1)^2} f(\frac{z}{1-z})dz$$

INFINITE RANGE INTERVALS

- While a common choice, there are many substitutions possible, some will work better than others. In general,

$$z = \frac{x}{c+x} \qquad z = \frac{x^\gamma}{1+x^\gamma} \qquad x = \tan z$$

- for any c or γ are common choices.
- The integral can also be broken up into pieces. For example an integral from minus infinity to infinity is the sum of two integrals one from minus infinity to zero and one from zero to infinity. Different substitutions can be done over the different ranges.
- Unfortunately there are no hard and fast rules. Trial and error may be necessary to find the best way to evaluate a given situation.

MULTIPLE INTEGRATION

- So far we have only discussed integration over one variable, but in physics we often have integrals over multiple variables. If we have an integral

$$\int_0^1 \int_0^1 f(x,y)dxdy$$

- one way to solve it is to define a function F(y) where

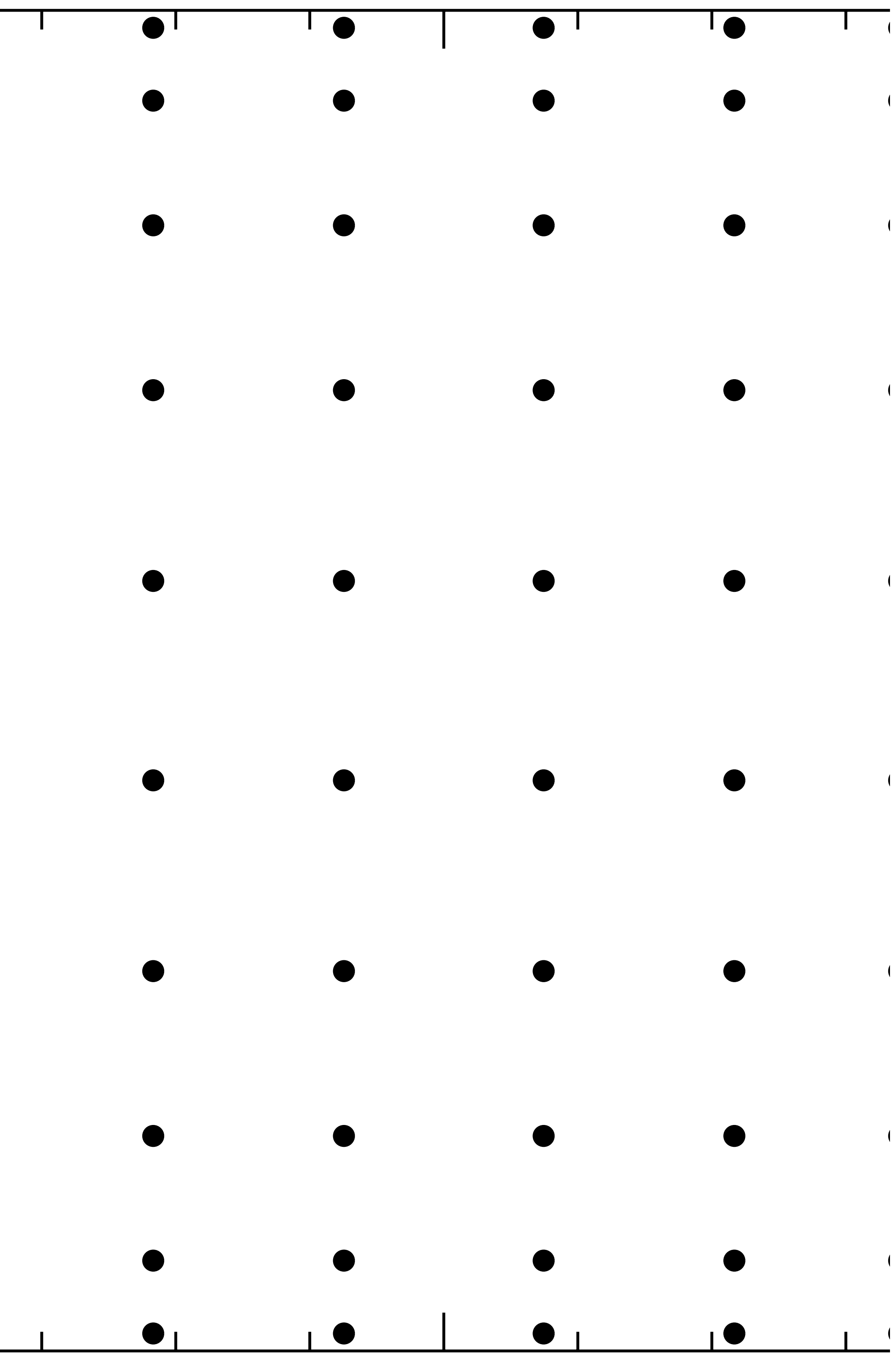
$$F(y) = \int_0^1 f(x,y)dx \qquad\qquad I = \int_0^1 F(y)dy$$

- So this way we would choose certain y values to use in evaluating our integral and then numerically determine those values by numerically integration over x to get F(y).

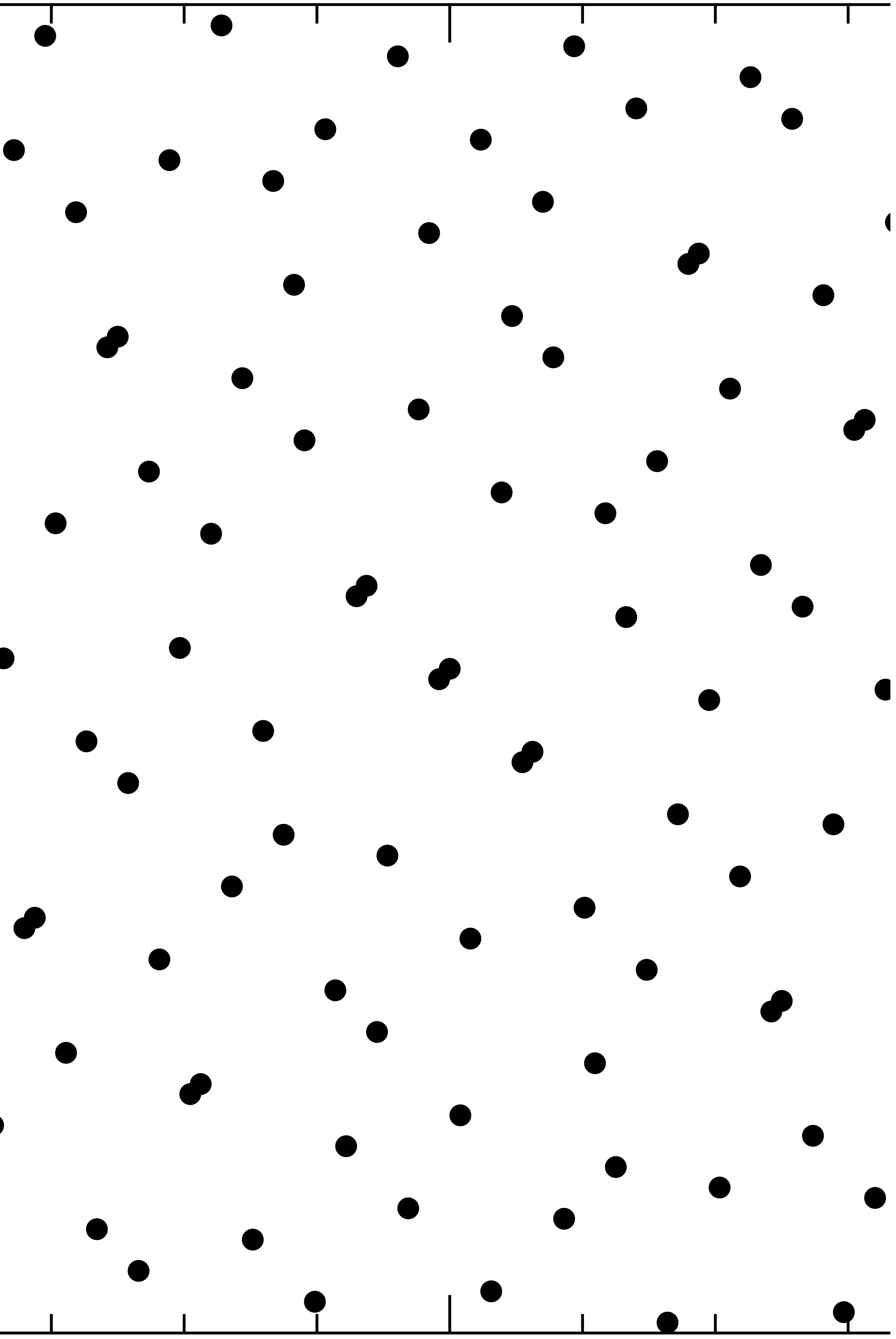
- Writing the sums we get the *Gauss-Legendre product formula*:

$$I \simeq \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j f(x_i, x_j)$$

MULTIPLE INTEGRATION

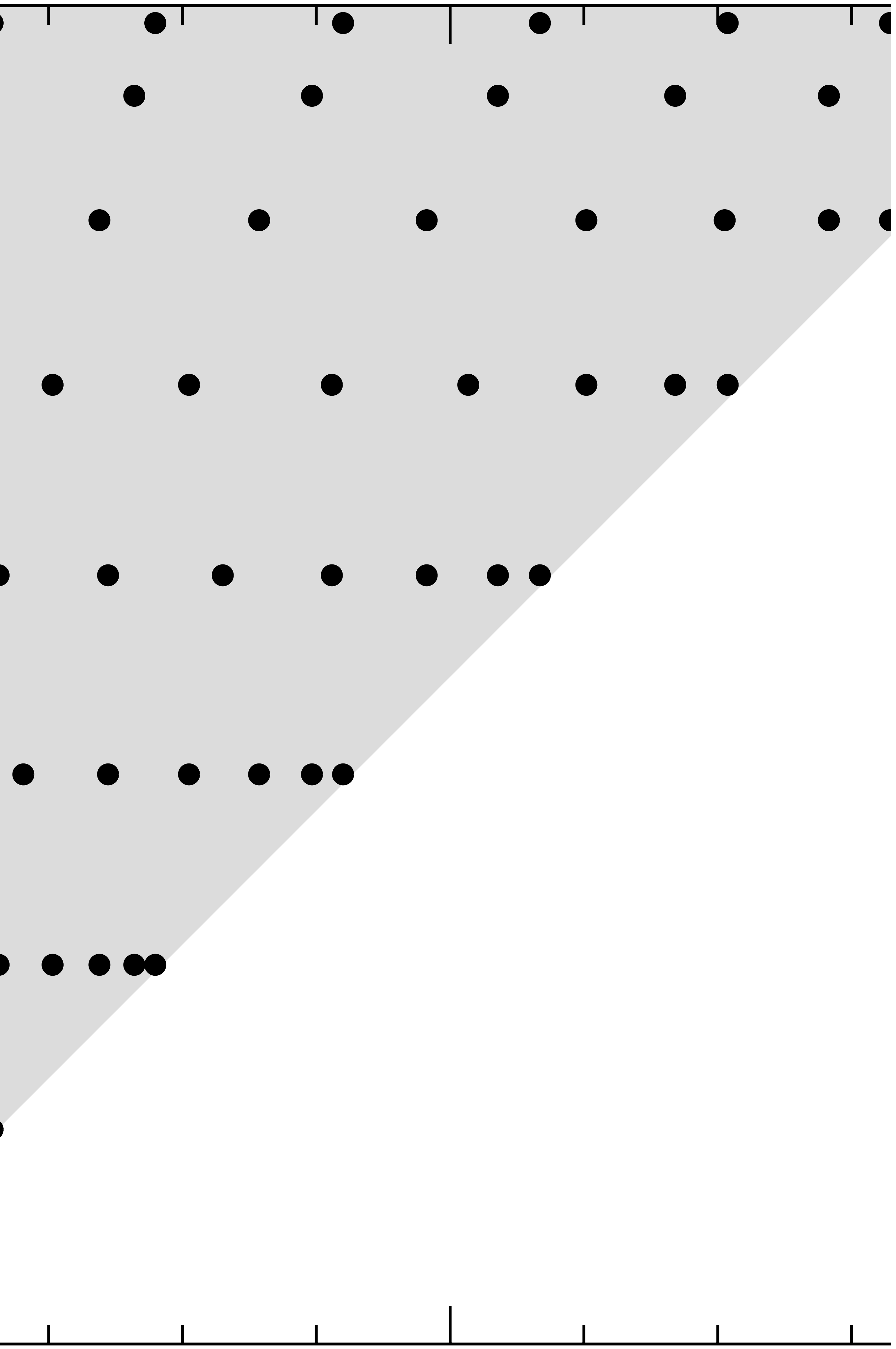


- If we were using quadrature we could end up with a point distribution like the following.
- However, there is no more reason to have an evenly spaced grid in 2D then there was to have evenly spaced steps in 1D.
- A number of schemes for choosing points in higher dimensions exist, but there is no known best solution.



MULTIPLE INTEGRATION

- The distribution to the left is called the Sobol sequence and is one choice that seems to give results in higher dimensions.
- Another approach is to just choose the points randomly. This is called Monte Carlo integration and we will discuss it later in the term.



MULTIPLE INTEGRATION

- We have been looking at integrals that are both to fixed numbers.
- But we also encounter integrals where the limit on one is a variable:

$$I = \int_0^1 dy \int_0^y dx\, f(x,y)$$

- Numerically this would lead to two sums, but where the second sum would not be to N, but instead to i as it steps up to N.
- This would give something like the points you see on the left, but it isn't clear this is a very good scheme as you can see some places have a lot of points while others have few.

MULTIPLE INTEGRATION

- In higher dimensions you can all kinds of strange volumes to integrate over.
- This can cause problems and there are various techniques to try and address them.
- One choice, as mentioned, is Monte Carlo integration where points are selected randomly.
- Others we won't go in to, but if you have a case of multiple integration over a volume that seems complicated be aware that straightforward application of a grid may not be a very efficient method for evaluating the integral.

SCIPY.INTEGRATE

- The scipy package has a sub-package integrate which contains most of the algorithms we have discussed and more.
- The methods quad (dblquad, tplquad) is a general function for integrating over 1(2,3) variables. Your limits can be infinite. The functions are called with quad(func,a,b) where a and b are your limit and func is the function you want to integrate over.
- The sub-package also includes quadrature() to do Guassian quadrature and romberg() for Romberg.
- Trapezoid() and Simpson() also exist, but they work on an array of values not a function. trapezoid(y, dx=1.0)

TERMINOLOGY

- Trapezoid Rule - Integration on equally spaced points where a line is drawn to interpolate the function between the points.
- Simpsons Rule - Integration on equally spaced points where a quadratic is used to interpolate between the points.
- Newton - Cotes - The general name for numerical integration on equally spaced points.
- Romberg Integration - A technique where lower number of points calculations of an integral are used to improve the accuracy of the calculation.
- Gaussian Quadrature - A method of numerical integration where points are chosen by the roots of Legendre polynomials.
- Approximation Error - The error introduced because the calculation is only approximately equal to the desired result.