Pricing via quadrature

Goals:

- Introduce the concept of numerical quadrature
- Outline construction of Gaussian nodes
- Review different quadrature rules

Related literature:

- Seydel, Appendix C
- en.wikipedia.org/wiki/Gaussian_quadrature

Integral representation

- Suppose that the random variable X has a probability density p such that

$$\mathbb{P}[X \leq x] = \int_{-\infty}^{x} p(u)\, du$$

- In this case, the expectation $\mathbb{E}[f(X)]$ can be rewritten as

$$\mathbb{E}[f(X)] = \int f(u)p(u)\, du$$

- If we know $p(\cdot)$, then we can evaluate $\mathbb{E}[f(X)]$ by computing the integral above.

Integral representation

- For some models of asset prices we indeed know p; e.g., Binomial, Black-Scholes or Merton models.
- For example, the log-normal density in Black-Scholes or Merton models is

$$p(u) = \frac{1}{u\sigma\sqrt{2\pi}} \exp\left(-\frac{(\log u - \mu)^2}{2\sigma^2}\right) \tag{1}$$

  This density is practically never used; people change variables so that the integrand is the normal density. I'm not even sure why I included it here.
- We are going to explore the quadrature approach in this lecture.

Evaluating integrals

- Consider an integral of the form

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

- Unless you know the antiderivative of $f(x)$, there is no closed form expression for the above integral. In finance, we almost never encounter functions whose antiderivatives are known.
- Therefore, we need to compute an approximation to the integral.
- This means that we must find weights $w_i \in \mathbb{R}$ and nodes $x_i \in [a, b]$ so that

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{N} w_i f(x_i) \tag{2}$$

Quadrature

- It would be really, really nice if the weights and the nodes did not depend on the particular function being integrated. Then they could be precomputed and stored.
- You will notice that all the quadrature schemes that we will see have this property.
- Of course, the number of nodes necessary to achieve a given precision might depend on the function or some quadrature methods might work better than another on a given function, but the nodes and the weights will never depend of the function explicitly.

Riemann sums

For example, if the weights $w_i = (b-a)/N$ for all i and the nodes

$$x_i = a + \frac{b-a}{N}(i-1) \qquad i = 1, \ldots, N \tag{3}$$

then the approximation is the left Riemann sum. (Note independence from f!)

For the right Riemann sums

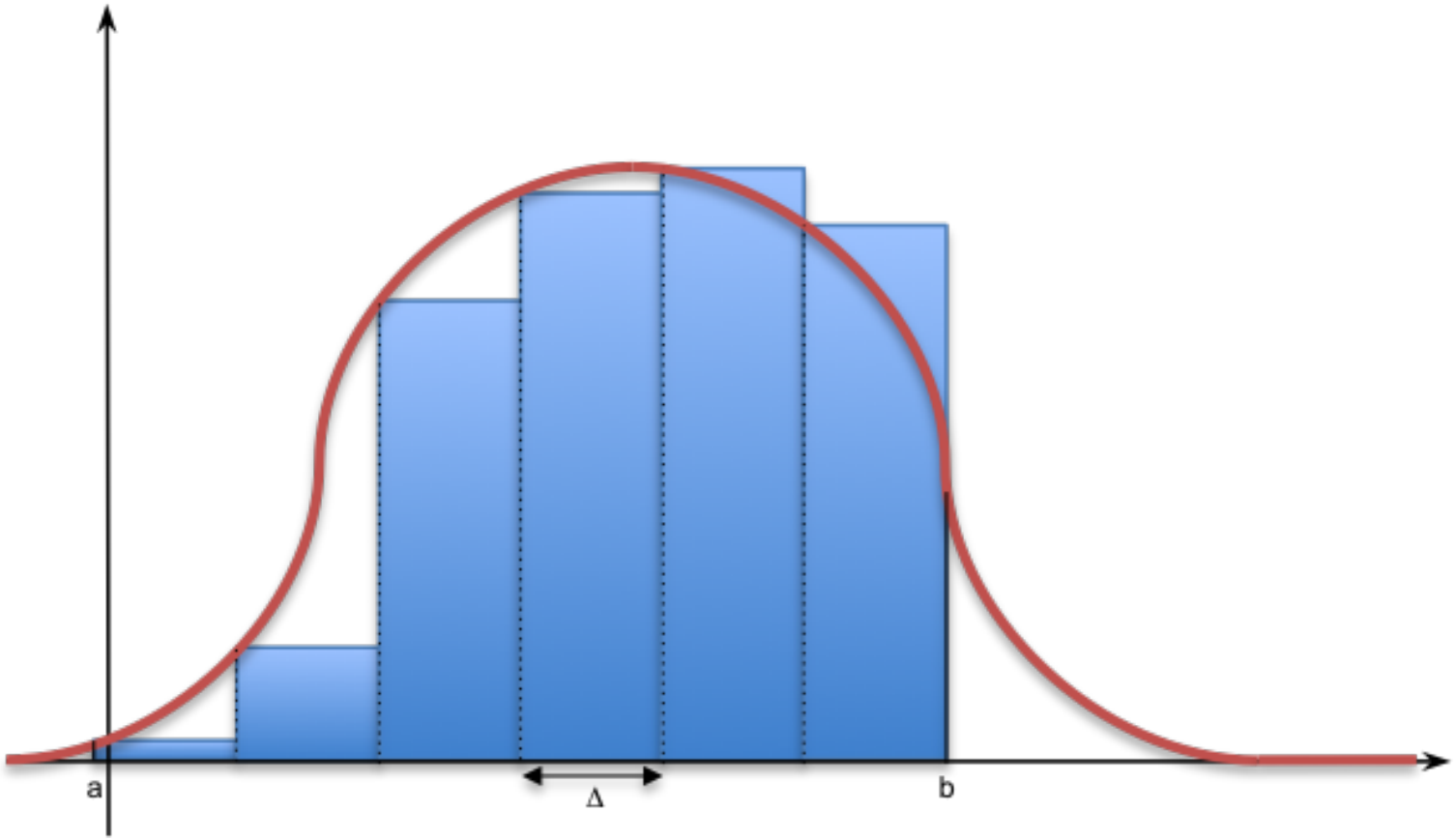
$$x_i = a + \frac{b-a}{N} i \qquad i = 1, \ldots, N \tag{4}$$

Which is better? To find out, let's pick $\epsilon \in [0, 1]$ and define

$$x_i = a + \frac{b-a}{N}(i-1) + \epsilon \frac{b-a}{N} \qquad i = 1, \ldots, N \tag{5}$$

If $\epsilon = 0$, (5) reduces to (3) and if $\epsilon = 1$, (5) reduces to (4).

Left-point Riemann sum



Rectangular quadrature

To find the best ϵ, let's warm up on the case when $N = 1$. Then there is only one node

$$x_1 = a + (b - a)\epsilon = a(1 - \epsilon) + b\epsilon \tag{6}$$

and only one weight $w_1 = b - a$.

Approximation (2) becomes

$$\int_a^b f(x)\,dx \approx (b - a)f(x_1) = (b - a)f(a(1 - \epsilon) + b\epsilon) \tag{7}$$

In other words, we are trying to approximate the integral with the value of the function at a single point. Note that this point does not depend on the function itself.

How good an approximation is it?

Accuracy of rectangular quadrature

To find out, let's Taylor expand the integrand $f(x)$ about the point x_1 and note (by Taylor's theorem) that

$$f(x) = f(x_1) + f'(x_1)(x - x_1) + \frac{1}{2} f''(\xi)(x - x_1)^2 \tag{8}$$

for some number $\xi \in [a, b]$. We are clearly assuming here that f is twice differentiable. In practice, you will find that the integrands are usually not differentiable, but are piecewise very differentiable, so you would need to break up the integration domain into pieces. (Imagine payoff of a call, for example, integrated against a smooth density.)

First note that because of (6)

$$b - x_1 = b - a(1 - \epsilon) - b\epsilon = (b - a)(1 - \epsilon) \tag{9}$$

and

$$a - x_1 = a - a(1 - \epsilon) - b\epsilon = -(b - a)\epsilon \tag{10}$$

Accuracy of rectangular quadrature

Using (8), we see that the error of the approximation (7) is

$$\int_a^b f(x)\,dx - (b-a)f(x_1) = f'(x_1)\int_a^b (x-x_1)\,dx + \frac{1}{2}\int_a^b f''(\xi)(x-x_1)^2\,dx \tag{11}$$

where the first term on the right-hand side (due to (9) and (10)) is

$$f'(x_1)\int_a^b (x-x_1)\,dx = \frac{1}{2}f'(x_1)(b-a)^2\left[(1-\epsilon)^2-\epsilon^2\right] \tag{12}$$

and the second is

$$\begin{aligned}\frac{1}{2}\int_a^b f''(\xi)(x-x_1)^2\,dx &\leq \frac{1}{2}\max_{\xi\in[a,b]}|f''(\xi)|\int_a^b (x-x_1)^2\,dx \\ &= \frac{1}{6}\max_{\xi\in[a,b]}|f''(\xi)|\,(b-a)^3\left[(1-\epsilon)^3+\epsilon^3\right]\end{aligned} \tag{13}$$

Accuracy of rectangular quadrature

For narrow intervals (when a and b are close) (13) is a lot smaller than (12). But we can do better still: remember ϵ is ours to choose! (As long as we don't make it depend on f.)

By setting ϵ to 0 and 1, we see that the left and right Riemann sums produce the same size error in (12), but if we set $\epsilon = 1/2$, then (12) disappears altogether, leaving (13) as the whole error bound, which equals

$$\frac{1}{24} \max_{\xi \in [a,b]} |f''(\xi)| \, (b-a)^3 = O((b-a)^3), \tag{14}$$

when $\epsilon = 1/2$.

- Therefore, for narrow intervals, (17) tells us the midpoint rule with the error $O((b-a)^3)$ is an order of magnitude more accurate than the left or right Riemann rules, which are $O((b-a)^2)$.

Accuracy of rectangular quadrature

The analysis on the last few slides applies to any interval. Therefore we can take the original interval $[a, b]$, cut it into N equal intervals and apply the analysis to each one. We will then end up with the weights $w_i = 1/N$ and the nodes (for the midpoint rule)

$$x_i = a + \frac{b-a}{N}\left(i - \frac{1}{2}\right) \qquad i = 1, \ldots, N \tag{15}$$

The error on each interval of length $(b-a)/N$ will be smaller than

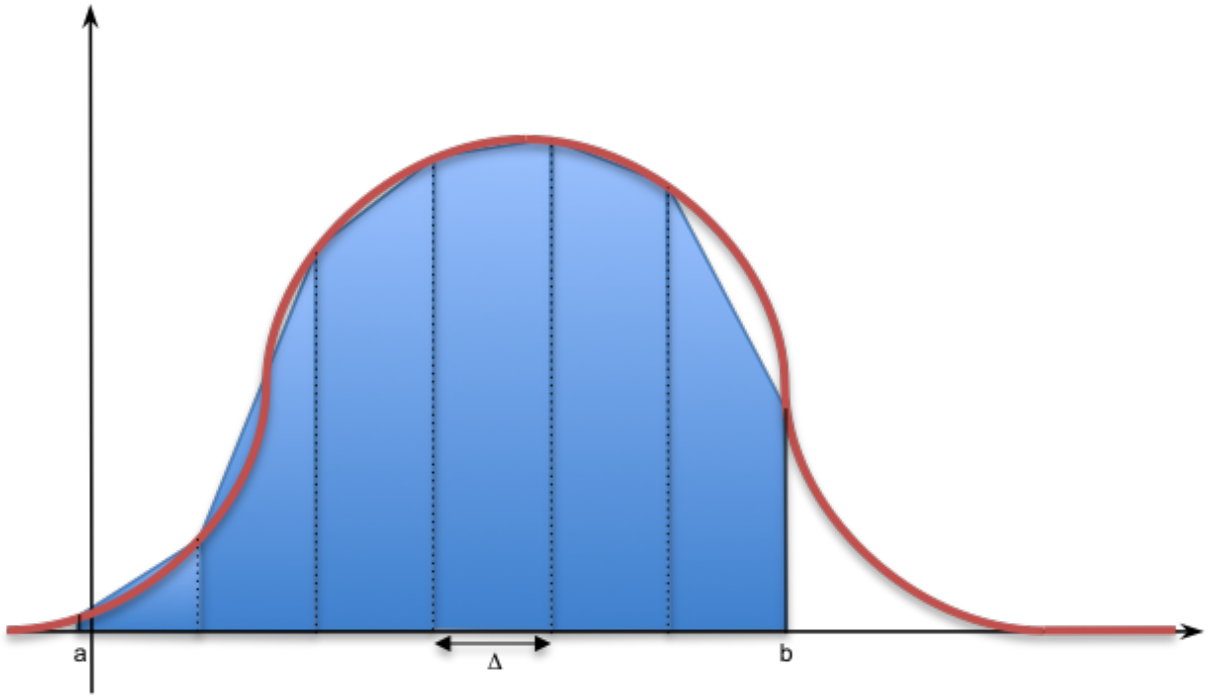
$$\frac{1}{24} \max_{\xi \in [a,b]} |f''(\xi)| \left(\frac{b-a}{N}\right)^3 \tag{16}$$

and, therefore, the total error of the quadrature on $[a, b]$ will be smaller than

$$\frac{1}{24} \max_{\xi \in [a,b]} |f''(\xi)| \, (b-a)^3 \frac{1}{N^2} \tag{17}$$

Trapezoidal quadrature

A close cousin of the midpoint rule is the trapezoidal rule:



If f is twice differentiable with bounded second derivative, the absolute error of the trapezoidal quadrature rule is also of order $O(N^{-2})$

Quadrature from another point of view

- Let's take another look at the single node approximation (7).
- If $\epsilon \neq 1/2$, then the approximation to the integral will be exact only for constant functions f, i.e., for polynomials of order 0.
- However, if we choose $\epsilon = 1/2$, then the approximation will be exact for all linear functions as well, i.e., for polynomials of order one. [Why?]
- In other words, by choosing a node cleverly we increase the class of functions that will be approximated *exactly*. Again, notice that the node does not depend on any particular function f. (OK, it's the last time I mention that.)
- Can this idea be generalized? Is it possible to choose weights and nodes so that the integrals of higher and higher order polynomials will be approximated exactly?

Quadrature from another point of view

One might imagine that since a polynomial of degree less than N is determined by its values at N points, there should be some kind of N-point quadrature rule that would produce exact results with that many nodes. Indeed, let x_i be an arbitrary collection of nodes such that

$$a \leq x_1 < \ldots < x_N \leq b \tag{18}$$

Our first goal is to find the weights w_i that would be independent of f and the approximation (2) would hold exactly for polynomials of degree less than N.

Lagrange Polynomials

To help with that consider the following polynomial:

$$L_i(x) = \prod_{\substack{1 \leq j \leq N \\ j \neq i}} \frac{x - x_j}{x_i - x_j} \tag{19}$$

For example,

$$L_1(x) = \frac{x - x_2}{x_1 - x_2} \cdots \frac{x - x_N}{x_1 - x_N} \tag{20}$$

It has degree $N - 1$ and

$$L_i(x_j) = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases} \tag{21}$$

So, $L_i(x)$ is a sort of a delta function on the nodes $\{x_j\}$.

Lagrange Polynomials

Polynomials of the form (19) are called *Lagrange polynomials*. As you can guess from their having a name, they are fairly useful.

Here is one such use: imagine that $f(x)$ is a polynomial of degree less than N. Then the following is true:

$$f(x) = \sum_{i=1}^{N} f(x_i) L_i(x) \tag{22}$$

Indeed, both the left and the right hand side are polynomials of degree less than N. But they are equal at all x_j for $j = 1, \ldots, N$. [Why?] Therefore, they are the same polynomial, albeit written differently.

Lagrange Polynomials and Quadrature

Why is this useful to us? Look at the definition (19) of $L_i(x)$ again. Do you see any mention of f? No! Therefore, they are independent of f. Now, let's integrate both sides of (22):

$$\int_a^b f(x)\,dx = \sum_{i=1}^{N} f(x_i) \int_a^b L_i(x)\,dx = \sum_{i=1}^{N} w_i f(x_i) \tag{23}$$

if we define the weights

$$w_i := \int_a^b L_i(x)\,dx \tag{24}$$

And, so, we just derived a quadrature rule that with N points will approximate all polynomials of degree less than N exactly.

Other functions can be approximated by their Taylor series and, therefore, their error will start with the Nth Taylor term.

Lagrange Polynomials and Quadrature

The quadrature rule (23) does not have a name.

Why? Because there is a far better one.

Notice first that the rule (23) holds for any and every collection of x_is in (18).

It stands to reason that by selecting the nodes judiciously one can do better.

The quadrature rule on the next slide tells us just how much better and it does earn a name and a pretty respectable one at that.

Gaussian nodes

Theorem 1. *For any positive integer N, there is a set of N nodes x_i and N weights w_i such that*

$$\int_{-1}^{1} f(x)\,dx = \sum_{i=1}^{N} w_i f(x_i) + R_N \tag{25}$$

where R_N is the remainder term:

$$R_N = \frac{2^{2N+1}(N!)^4}{(2N+1)[(2N)!]^3} f^{(2N)}(\xi) \tag{26}$$

for some $\xi \in [-1, 1]$.

In particular, the quadrature is *exact* whenever f is a polynomial of degree less than $2N$.

Stirling's asymptotic formula tells us that $R_N = O(N^{-2N})$ as $N \to \infty$ if $f^{(2N)}$ doesn't grow exponentially with N.

Gaussian nodes

How can something like (33) be true?

To find out, we need an excursion into orthogonal polynomials. The vector space $\mathbb{P}^{N-1}$ of polynomials on the interval $[-1, 1]$ of degree less than N is spanned by the polynomials $\{1, x, x^2, \ldots, x^{N-1}\}$. [a] We define a dot product (also called inner product) of any two polynomials p and q on this vector space (thus making it a Hilbert space) by

$$\langle p, q \rangle := \int_{-1}^{1} p(x)q(x)\,dx \tag{27}$$

Under this inner product, using Gram-Schmidt for example, we can come up with an orthonormal basis of $\mathbb{P}^{N-1}$. We will denote the elements of this basis by $\{p_0(x), p_1(x), \ldots, p_{N-1}(x)\}$.

[a] Note that $\mathbb{P}^{N-1}$ has nothing to do with the probability measure $\mathbb{P}$.

Gaussian nodes

Now we can get on with the proof of the theorem on slide 20.

Let $f(x)$ be a given polynomial of degree less than $2N$ and let us choose a non-zero polynomial $p_N(x)$ of degree N orthogonal to all elements of the basis of $\mathbb{P}^{N-1}$, and, therefore, to all polynomials in $\mathbb{P}^{N-1}$.

Then we can find polynomials $q(x)$ and $r(x)$ of degree less than N so that

$$f(x) = p_N(x)q(x) + r(x). \tag{28}$$

Here $q(x)$ is the quotient and $r(x)$ is the remainder after dividing $f(x)$ by $p_N(x)$. Since $q \in \mathbb{P}^{N-1}$ and, thus, orthogonal to $p_N(x)$, it follows that

$$\int_{-1}^{1} f(x)\,dx = \int_{-1}^{1} p_N(x)q(x)\,dx + \int_{-1}^{1} r(x)\,dx = \int_{-1}^{1} r(x)\,dx. \tag{29}$$

Gaussian nodes

- Note that in (29) $r(x)$ depends on $f(x)$, but not on the choice of the N nodes and $p_N(x)$ depends neither on $f(x)$ nor on the nodes.
- Moreover, we are free to choose $p_N(x)$ so long as its degree is N and it's orthogonal to $\mathbb{P}^{N-1}$. Therefore, we are free to pick the nodes as we wish and (29) will still hold regardless of our choice.

Gaussian nodes

Here is the crux:

- Let's pick the N quadrature nodes (18) to be the roots of $p_N(x)$.
- We will prove later—on slide 29—that these N roots are all distinct and lie in $(-1, 1)$.
- Define the weights w_i by (24) with $L_i(x)$, where we set the nodes $\{x_i\}$ to be the roots of $p_N(x)$.

Gaussian nodes

Let's apply this quadrature rule to f and see what happens:

$$\sum_{i=1}^{N} w_i f(x_i) = \sum_{i=1}^{N} w_i p_N(x_i) q(x_i) + \sum_{i=1}^{N} w_i r(x_i) = \sum_{i=1}^{N} w_i r(x_i) \quad (30)$$

because x_is are the roots of p_N. And since degree of $r(x)$ is less than N, quadrature rule (23) tells us that

$$\sum_{i=1}^{N} w_i r(x_i) = \int_{-1}^{1} r(x)\, dx. \quad (31)$$

because it applies to all polynomials of degree less than N regardless of the choice of the N nodes.

Gaussian nodes

Combining (30) and (31) with (29), we see that if f is a polynomial of degree less than $2N$, we get the promised result:

$$\int_{-1}^{1} f(x)\,dx = \sum_{i=1}^{N} w_i f(x_i) \tag{32}$$

if we choose the N nodes to be the roots of p_N.

The error estimate (34) follows from Taylor theorem (with some effort).

Generalized Gaussian Quadrature

We can generalize Gaussian Quadrature by allowing integration against any (positive) weight function $w(x)$ over a (possibly infinite) interval $\mathbb{D}$.

For example, in the discussion so far, $\mathbb{D}$ was the interval $[-1, 1]$ and $w(x)$ was identically one.

For the sake of completeness, we include the statement of the more general theorem on the next slide.

Generalized Gaussian Quadrature

Theorem 2. *Let $w(x)$ be a positive function on an interval $\mathbb{D}$. For any positive integer N, there is a set of N nodes x_i and N weights w_i such that*

$$\int_{\mathbb{D}} f(x)w(x)\,dx = \sum_{i=1}^{N} w_i f(x_i) + R_N. \tag{33}$$

where R_N is the remainder term:

$$R_N = c(N)f^{(2N)}(\xi) \tag{34}$$

for some constant c that depends only on N and some $\xi \in \mathbb{D}$. The weights w_i are given by

$$w_i = \int_{\mathbb{D}} L_i(x)w(x)\,dx \tag{35}$$

In particular, the quadrature is *exact* when f is a polynomial of degree less than $2N$.

Roots of $p_N(x)$

We only have one thing left to prove: that all the roots of $p_N(x)$ lie in the interval $\mathbb{D}$, which so far has been $[-1, 1]$:

Lemma 1. *Let $p_N(x)$ be a polynomial of degree N orthogonal to $\mathbb{P}^{N-1}$, i.e., to all polynomials of degree less than N. Then $p_N(x)$ must have N distinct roots in $\mathbb{D}$.*

Proof. Let the points $c_1, \ldots, c_M$ be all the points in $\mathbb{D}$ where $p_N(x)$ changes sign. Each c_i is a root of $p_N(x)$, and therefore, $M \leq N$.

Now let us stare at the polynomial

$$p_N(x) \prod_{i=1}^{M} (x - c_i). \tag{36}$$

It does not change sign at any of the c_i [Why?] and, therefore, does not change sign at all on $\mathbb{D}$.

Therefore, it is always either non-negative or non-positive on the whole $\mathbb{D}$ and, being a non-trivial polynomial, it is not identically zero.
Therefore, since $w(x) > 0 \quad \forall x \in \mathbb{D}$,

$$\int_{\mathbb{D}} p_N(x) \left(\prod_{i=1}^{M} (x - c_i) \right) w(x)\, dx \neq 0. \tag{37}$$

Remember that $M \leq N$.

If $M < N$, then the polynomial $\prod_{i=1}^{M}(x - c_i)$ has degree less than N and, is, therefore, orthogonal to $p_N(x)$, which means that

$$\int_{\mathbb{D}} p_N(x) \left(\prod_{i=1}^{M} (x - c_i) \right) w(x)\, dx = 0 \tag{38}$$

which contradicts (37). Therefore, $M = N$, which means that $p_N(x)$ changes sign N times in $\mathbb{D}$, and, so it must have N roots in $\mathbb{D}$. $\square$

Examples of Gaussian quadratures with weights $w(x)$ and intervals $\mathbb{D}$

Name	$w(x)$	$\mathbb{D}$
Gauss-Legendre	1	$[-1, 1]$
Gauss-Laguerre	e^{-x}	$[0, \infty)$
Gauss-Hermite	e^{-x^2}	$(-\infty, \infty)$
Gauss-Jacobi	$(x+1)^a(1-x)^b$	$[-1, 1]$

For Gauss-Jacobi, there is a requirement that $a > -1$ and $b > -1$.

Multiple Dimensions

Often the instruments that we need to price depend on more than one variable. You will see one example in your homework: a option on one underlying instrument, but with the payment of the option contingent on what happens to another instrument.

In these situations we are forced to compute integrals of more than one variable.

There is no equivalent of Gaussian quadrature for multiple dimensions, but one can apply it to iterated integrals.

Multiple Dimensions

Suppose we are trying to compute an integral of the form

$$\int_a^b \int_c^x f(x,y)\,dydx = \int_a^b \left(\int_c^x f(x,y)\,dy \right) dx. \tag{39}$$

Let's say we use midpoint rule on the outer integral with N points. That will give us an error of $O(N^{-2})$ with $O(N)$ evaluations of the inner integral. But each evaluation of the inner integral will itself cost is $O(N)$. Therefore, $O(N^2)$ evaluations will give us $O(N^{-2})$ error. (Actually, the error will be even worse, but let's stay optimistic.)

If the integral is 3D, then we'll need $O(N^3)$ evaluations for the same error. And N can easily be a few hundred, so you can see what people mean when they refer to "The Curse of Dimensionality."

Therefore, in higher dimensions we will need other techniques.