Orthogonal Polynomials

Orthogonal polynomials arise from series solutions to differential equations, although they can be arrived at in a variety of different manners. Orthogonal polynomials are well studied, and their properties are generally well understood, so they are a useful tool, especially when used as a basis set.

The set of functions $\{\phi_0(x), \ldots, \phi_n(x)\}$ is **linearly independent** on $[a, b]$ if whenever

$$c_0\phi_0(x) + c_1\phi_1(x) + \cdots + c_n\phi_n(x) = \sum_{i=1}^{n} c_i\phi_i(x) = 0 \text{ for all } x \in [a, b]$$

then $c_0 = c_1 = \cdots = c_n = 0$. Otherwise, the set is linearly dependent.

An integrable function w is called a **weight function** on $[a, b]$ if $w(x) \geq 0$ for $x \in [a, b]$ but $w(x) \not\equiv 0$ on any subinterval of $[a, b]$. The weight function assigns varying degrees of importance to portions of the interval $[a, b]$.

The set of functions $\{\phi_0(x), \ldots, \phi_n(x)\}$ is **orthogonal** on $[a, b]$ with respect to the weight function w if

$$\int_a^b w(x)\phi_i(x)\phi_j(x)\,dx = \alpha_i\delta_{ij}$$

where $\delta_{ij} = 1$ if $i = j$ and is zero otherwise. The constant $\alpha_i > 0$.

If $\alpha_i = 1$ for all i, then the set of functions is **orthonormal**.

Example For a given positive integer n, the set of functions $\{\phi_0(x), \ldots, \phi_{2n-1}(x)\}$

$$\begin{aligned}
\phi_0(x) &= \frac{1}{\sqrt{2\pi}}, \\
\phi_k(x) &= \frac{1}{\sqrt{\pi}}\cos kx, \text{ where } k = 1, 2, \ldots, n, \\
\phi_{n+k}(x) &= \frac{1}{\sqrt{\pi}}\sin kx, \text{ where } k = 1, 2, \ldots, n-1,
\end{aligned}$$

is an orthonormal set on interval $[-\pi, \pi]$ with respect to weight function $w(x) = 1$.

This set of functions can be used as a basis set to create a least squares approximation to any function f in the interval $[-\pi, \pi]$

$$f(x) \sim S_n(x) = \sum_{k=0}^{2n-1} a_k\phi_k(x),$$

where

$$a_k = \int_{-\pi}^{\pi} f(x)\phi_k(x)\,dx.$$

The function $\lim_{n\to\infty} S_n(x)$ is called the Fourier series of f.

The basic idea is that for a given weight function and interval we can create a set of orthogonal polynomials using Gram Schmidt orthogonalization.

Here is a table of common orthogonal polynomials.

Polynomial $\phi_n(x)$		Interval (a,b)	weight $w(x)$	Normalization $\int_a^b w(x)\phi_n^2(x)\,dx$
Legendre	$P_n(x)$	$(-1,1)$	1	$2/(2n+1)$
Laguerre	$L_n(x)$	$(0,\infty)$	e^{-x}	1
Hermite	$H_n(x)$	$(-\infty,\infty)$	e^{-x^2}	$2^n n!\sqrt{\pi}$
Chebyshev 1st Kind	$T_n(x)$	$(-1,1)$	$(1-x^2)^{-1/2}$	$\pi(n=0)$ or $\pi/2(n\neq 0)$
Chebyshev 2nd Kind	$U_n(x)$	$(-1,1)$	$(1-x^2)^{1/2}$	$\pi/2$

Note we could include the endpoints on the intervals, $(-1,1)$ or $[-1,1]$, etc.

The Gram-Schmidt Process to Construct Orthogonal Polynomials

The set of polynomials $\{\phi_0(x),\ldots,\phi_n(x)\}$ defined in the following way is orthogonal on $[a,b]$ with respect to the weight function w.

$$\phi_0(x) = 1,$$

$$B_1 = \frac{\displaystyle\int_a^b x\,w(x)\,[\phi_0(x)]^2\,dx}{\displaystyle\int_a^b w(x)\,[\phi_0(x)]^2\,dx},$$

$$\phi_1(x) = x - B_1,$$

and then for $k \geq 2$ use

$$\phi_k(x) = (x - B_k)\phi_{k-1}(x) - C_k\phi_{k-2}(x),$$

$$B_k = \frac{\displaystyle\int_a^b x\,w(x)\,[\phi_{k-1}(x)]^2\,dx}{\displaystyle\int_a^b w(x)\,[\phi_{k-1}(x)]^2\,dx},$$

$$C_k = \frac{\displaystyle\int_a^b x\,w(x)\,\phi_{k-1}(x)\phi_{k-2}(x)\,dx}{\displaystyle\int_a^b w(x)\,[\phi_{k-2}(x)]^2\,dx}.$$

You can prove this using induction.

The *Mathematica* file shows how this can be used to construct the Legendre polynomials mentioned earlier.

Theorem If the set of orthogonal polynomials $\{\phi_0(x),\ldots,\phi_k(x),\ldots,\phi_n(x)\}$ is defined on $[a,b]$ with weight function w, and $\phi_k(x)$ is a polynomial of degree k then $\phi_k(x)$ has k distinct roots (when $k \geq 1$) in the interval (a,b).

Gaussian Quadrature: Initial Thoughts

1. Assume we can approximate the integral as the following sum:

$$\int_a^b w(x)f(x)\,dx \sim \sum_{i=1}^{n} c_i f(x_i),$$

where we will have nodes $x_1, \ldots, x_n \in [a, b]$.

2. We want to find these nodes in an optimal way, rather than just having them equally spaced.

3. Our earlier theorem tells us the orthogonal polynomial $\phi_n(x)$ we define through the Gram-Schmidt process will have n roots in the interval.

4. We have $2n$ parameters to determine: c_i and x_i for $i = 1, \ldots, n$.

5. A polynomial of degree $2n-1$ has $2n$ parameters:

$$a_0 + a_1 x + \cdots + a_{2n-1}x^{2n-1} = \sum_{i=0}^{2n-1} a_i x^i$$

Key Point: If $f(x)$ is a polynomial of degree $2n-1$ or less, then we should get an exact result for the integral since we have enough parameters to fit the curve exactly!

A quadrature rule with **degree of precision** k means the quadrature rule gives the exact result for any polynomial of degree less than or equal to k

Example If $n = 3$ and $w(x) = 1$ and $[a, b] = [-1, 1]$, our approximation should be exact if $f(x) = 1, x, x^2, x^3, x^4, x^5$.

Choose c_i and x_i such that

$$\int_{-1}^{1} x^k\,dx = \sum_{i=1}^{3} c_i x_i^k, \;\; k = 0, 1, \ldots, 5.$$

This is a set of 6 nonlinear equations in 6 unknowns, which we will use *Mathematica* to solve. We get

$c_1 = 0.555555$
$c_2 = 0.555555$
$c_3 = 0.888888$
$x_1 = 0.774597$
$x_2 = -0.774597$
$x_3 = 0$

Each c_i is the weight associated with the node x_i.

Now that we have these parameters, we can use the formula

$$\int_{-1}^{1} f(x)\,dx \sim \sum_{i=1}^{n} c_i f(x_i),$$

for any other function f we like.

In this case, since $w(x) = 1$ and $[a, b] = [-1, 1]$ and $n = 3$, the nodes are the roots of the Legendre polynomial $P_3(x)$.

Proof of Gaussian Quadrature Technique for $[a, b] = [-1, 1]$ and $w(x) = 1$ case

Now that we have an idea of what to expect, let's work out the details for general n, and show that Gaussian quadrature has **degree of precision** $2n - 1$.

1. We again begin with the Lagrange interpolating polynomial.

$$f(x) = \sum_{i=1}^{n} f(x_i) \prod_{k=1, k\neq i}^{n} \frac{(x - x_k)}{(x_i - x_k)} + \frac{1}{n!} f^{(n)}(c(x)) \prod_{k=1}^{n} (x - x_k). \tag{1}$$

   We will use the following quadrature rule, which is found by integrating Eq. (1). Since the error in Eq. (1) involves $f^{(n)}$, the quadrature rule will be have degree of precision at least $n - 1$.

$$\int_{-1}^{1} f(x)\, dx = \sum_{i=1}^{n} c_i f(x_i), \tag{2}$$
$$\text{where } c_i = \int_{-1}^{1} \prod_{k=1, k\neq i}^{n} \frac{(x - x_k)}{(x_i - x_k)}\, dx.$$

2. Suppose W is a any polynomial with degree $k \leq 2n - 1$ (W is not necessarily a Legendre polynomial). Let's show the quadrature rule Eq. (2) is exact for $f = W$.

   (a) Dividing W by the nth degree Lengendre polynomial gives

$$W(x) = Q(x)P_n(x) + R(x), \tag{3}$$

   where both Q and R are polynomials of degree less than n.

   (b) Since Q is degree less than n, Q can be written in terms of the Legendre polynomials as:

$$Q(x) = \sum_{i=0}^{n-1} d_i P_i(x)$$

   since the Legendre polynomials $\{P_0(x), \ldots, P_{n-1}(x)\}$ form a basis set for polynomials of degree at most $n - 1$.

   (c) Using the orthogonality property of the Legendre polynomials, we have

$$\int_{-1}^{1} Q(x)P_n(x)\, dx = \sum_{i=0}^{n-1} d_i \int_{-1}^{1} P_i(x)P_n(x)\, dx = 0,$$

   and, upon integrating Eq. (3),

$$\int_{-1}^{1} W(x)\, dx = 0 + \int_{-1}^{1} R(x)\, dx = \sum_{i=1}^{n} c_i R(x_i), \tag{4}$$

   where the last equality follows by using the quadrature rule (2) (with degree of precision $n - 1$) on the polynomial $R(x)$ which has degree at most $n - 1$.

(d) If we choose x_i, $i = 1, 2, \ldots, n$, to be the roots of the Lengendre polynomial $P_n(x)$, then Eq. (3) tells us that

$$W(x_i) = Q(x_i)\cancelto{0}{P_n(x_i)} + R(x_i) = R(x_i),$$

and therefore, we have from (4)

$$\int_{-1}^{1} W(x)\,dx = \sum_{i=1}^{n} c_i W(x_i),$$

so the quadrature rule Eq. (2) is exact for the any polynomial W of degree $2n-1$.

This analysis shows us why the nodes x_i should be chosen as the roots of the Legendre polynomial, and also gives us a formula to compute the weights,

$$c_i = \int_{-1}^{1} \prod_{k=1, k\neq i}^{n} \frac{(x - x_k)}{(x_i - x_k)}\,dx.$$

The nice thing is that for a variety of n the nodes and weights have already been calculated, so you don't have to work them out yourself!

The amazing connection to orthogonal polynomials is that the nodes are the roots of the orthogonal polynomial $\phi_n(x)$ we define through the Gram-Schmidt process on interval $[a, b]$ with weight function $w(x)$!

Notice the text defined Gaussian quadrature only for $[a, b] = [-1, 1]$ and $w(x) = 1$, but these ideas apply to the other orthogonal polynomials as well.

Gaussian Quadrature Technique for $[a, b] \neq [-1, 1]$ and $w(x) = 1$

If the finite interval $[a, b]$ is not $[-1, 1]$, use the linear transformation $t = (2x - b - a)/(b - a)$:

$$\int_a^b w(x) f(x)\,dx = \int_{-1}^{1} w\left(\frac{(b-a)t + b + a}{2}\right) f\left(\frac{(b-a)t + b + a}{2}\right) \cdot \frac{b-a}{2}\,dt$$

To Perform Gausssian Quadrature to Evaluate $\int_a^b w(x)f(x)\,dx$

$$L_{ni} = \prod_{k=1,k\neq i}^{n} \frac{(x-x_k)}{(x_i-x_k)}$$

I prefer to think of how it is built, so I get the weight in the correct place.

$$\begin{aligned} f(x) &= \sum_{i=1}^{n} f(x_i)L_{ni}, \\ \int_a^b w(x)f(x)\,dx &= \sum_{i=1}^{n} w_i f(x_i), \\ w_i &= \int_a^b w(x)L_{ni}(x)\,dx = \int_a^b w(x) \prod_{k=1,k\neq i}^{n} \frac{(x-x_k)}{(x_i-x_k)}\,dx \end{aligned}$$

You will sometimes see tables that reports weights based on the following, so be sure to read your tables closely!

$$\begin{aligned} f(x) &= \sum_{i=1}^{n} f(x_i)L_{in}, \\ \int_a^b f(x)\,dx &= \sum_{i=1}^{n} w_i f(x_i), \\ w_i &= \int_a^b L_{ni}(x)\,dx = \int_a^b \prod_{k=1,k\neq i}^{n} \frac{(x-x_k)}{(x_i-x_k)}\,dx \end{aligned}$$

You can use the formulas when the appropriate weight function is not present, but the results will not be as good.

Example $\int_0^\infty e^{-x}\cos x\,dx$

Identify the interval $[0,\infty)$ and $w(x) = e^{-x}$ as the weight function associated with Laguerre polynomials, $L_n(x)$. We choose the nodes to be roots of $L_n(x)$ and the weights can either be computed or looked up in a table. The function is $f(x) = \cos x$.

$$\begin{aligned} \cos(x) &= \sum_{i=1}^{5} \cos(x_i)L_{5i}, \\ \int_0^\infty e^{-x}\cos x\,dx &= \sum_{i=1}^{5} w_i \cos(x_i), \\ w_i &= \int_0^\infty e^{-x} \prod_{k=1,k\neq i}^{5} \frac{(x-x_k)}{(x_i-x_k)}\,dx \end{aligned}$$

The real power of the method is that the weights/nodes have already been tabulated and you can just plug them in!