Gaussian Quadrature

Justin Boyer

April 20, 2012

Introduction

Gaussian quadrature is an amazing technique for numerical integration. Generally when interpolating polynomials one needs $n+1$ points to interpolate an n^{th} degree polynomial. The same could be expected for numerical integration. However Gaussian quadrature integrates polynomials of degree $2n+1$ exactly with $n+1$ points. This is a very powerful technique.

Quadratures A numerical quadrature is a scheme, which is used to approximate the integral of a given function over a finite interval. The following is known as the Newton-Cotes formula, the right hand side is an approximation of the the integral on the left hand side, where the a_i are coefficients, and x_i are roots:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} a_i f(x_i), \ \ x_i \in [a, b]. \tag{1}$$

Different formulas give different a_i, the Newton-Cotes formulas chooses the x_i evenly. Generally this is a straight-forward approach and easily understandable and implementable. On large intervals, the Newton-Cotes formulas have an error of order h^k where h is step size of x_i and k is greater than one and depends on the method used. Another quadrature is Gaussian quadrature, which chooses x_i and a_i such that the error is minimized. This choice of x_i is made by finding the roots of an n^{th}-order Legendre polynomial $P_n(x)$. The family of Legendre polynomials are a collection of polynomials, $\{P_0(x), P_1(x), \dots, P_n(x), \dots\}$, where the polynomials are monic and orthogonal to each other with respect to $\omega(x) = 1$. Orthogonality in this context is with respect to the L_2 inner product, i.e.,

$$\int_{-1}^{1} P_n(x)P_m(x)dx = 0$$

except when $n = m$.

Gaussian Quadrature chooses x_i, and c_i for $i = 1, 2, \dots, n$, such that:

$$\int_{-1}^{1} P(x)dx \approx \sum_{i=1}^{n} c_i P(x_i), \ \ x_i \in [-1, 1]. \tag{2}$$

The integral on the left hand side is equal to the sum on the right hand side for any polynomial, $P(x)$, of degree strictly less than $2n$.[1]

[1] Burden-Faires Numerical Analysis 8th ed., pg 223

How do we find the polynomials such that the above equation holds. For a polynomial of degree 0 we use the easiest such polynomial, $P_0(x) = 1$. Likewise for a polynomial of degree 1 we use $P_1(x) = x$. These polynomials are orthogonal and monic, however for polynomials where the degree is greater than two this will not work so conveniently, since we want the polynomials to be monic and orthogonal. This is where Gram-Schmidt orthogonalization comes in handy. We can use Gram-Scmidt to generalize the process for polynomials of degree $n > 2$, which constructs an orthonormal basis of functions. The Legendre polynomials, $P_0(x), P_1(x), \ldots$, are one such set of linearly independent functions. The points needed to give exact results (for a polynomial of degree less than $2n-1$) for equation (2) are the roots of an n^{th} degree Legendre polynomial. Likewise the weights, c_i, necessary for the right hand side of equation (2) come from the coefficients of the Legendre polynomials.

Formulation

The first step towards generating the roots, x_i, and weights, c_i, for Gaussian Quadrature is to construct the corresponding polynomials. By applying Gram-Schmidt orthogonalization to the polynomials $1, x^2, x^3, \cdots$, on the interval $[-1, 1]$ we construct the Legendre polynomials, which we can then use to find the roots and the coefficients. Gram-Schmidt orthogonalization uses the fact that the L_2 inner product of two different degree polynomials will be zero, but the L_2 inner product of two polynomials with the same degree will be one. Given this concept of the Gram-Schmidt orthogonalization, the actual calculation is given by a three term recursion formula, i.e., the i^{th} polynomial (denoted ϕ_i) may be written in terms of the $(i-1)^{\text{th}}$ and the $(i-2)^{\text{th}}$ polynomials.

$$\phi_i(x) = (x - B_i)\phi_{i-1}(x) - D_i\phi_{i-2}(x) \tag{3}$$

where $B_i = \frac{\int_{-1}^{1} x[\phi_{i-1}(x)]^2 dx}{\int_{-1}^{1} [\phi_{i-1}(x)]^2 dx}$ and $D_i = \frac{\int_{-1}^{1} x[\phi_{i-1}(x)\phi_{i-2}(x)]dx}{\int_{-1}^{1} [\phi_{i-2}(x)]^2 dx}$

This formula may look different than the Gram-Schmidt formula we are familiar with for vectors. But it is not, for vectors we have:

$$u_i = v_i - \sum_{l=1}^{i-1} \frac{< v_i, u_l >}{< u_l, u_l >} u_l \tag{4}$$

where $< v_i, u_l >$ denotes the inner product between v_i and u_l. In order to see that the idea is the same, think of $x\phi_i$ as the vector we are projecting, let's call it v_i. Then we have

$$u_i = v_i - \frac{< v_i, v_{i-1} >}{< v_{i-1}, v_{i-1} >} v_{i-1} - \frac{< v_i, v_{i-2} >}{< v_{i-2}, v_{i-2} >} v_{i-2} \tag{5}$$

which is our familiar Gram-Schmidt orthogonalization and is the same as (3) if we note that $u_i = \phi_i$. Additionally (3) is only a three term recursion, because if we carry out more terms all of the them will be zero since two different degree polynomials are orthogonal. Thus using Gram-Schmidt orthogonalization we build a family of orthogonal, monic polynomials. The next step in Gaussian quadrature is to then compute the roots, x_i, of these polynomials. This can be done relatively easy using Newton's Method. Once the roots have been calculated the weights c_i are found using the roots and Lagrange polynomials. We then have a quadrature that is exact for polynomials of degree $2n + 1$.

Another interesting linear algebra tidbit regarding Legendre polynomials is the fact that the Legendre differential equation is:

$$\frac{d}{dx}[(1 - x^2)\frac{d}{dx}P(x)] = -\lambda P(x) \tag{6}$$

which is just an eigenvalue problem. It turns out that the Legendre polynomials are the eigenfunctions of this particular operator.

Example

In this section we will demonstrate an example in which we find the Legendre polynomials up to degree 3, then use those to calculate the integral of the polynomial of degree 5 exactly, e.g., $\int_{-1}^{1} 5x^5 - 7x^3 + x + 4dx$ will be calculated exactly using a polynomial of degree 3.

First we start with the easiest polynomial, $P_0(x) = 1$. We apply G-S (Gram-Schmidt orthogonalization). $P_1(x) = x - \frac{<x,1>}{<1,1>} = x - \frac{\int_{-1}^{1} x*1^2 dx}{\int_{-1}^{1} 1dx}$, which we swiftly calculate to be, $P_1(x) = x$. Next is, $P_2(x)$, which is not so straightforward, but still not too bad, we're just integrating polynomials, using (G-S) $P_2(x) = x^2 - \frac{<x^2,x>}{<x,x>}x - \frac{<x^2,1>}{<1,1>} = x^2 - \frac{\int_{-1}^{1} x^2 * x dx}{\int_{-1}^{1} x^2 dx}x - \frac{\int_{-1}^{1} x^2 * 1 dx}{\int_{-1}^{1} 1dx}$. Computing the necessary integrals we arrive at $P_2(x) = x^2 - 1/3$. Following the same blueprint we arrive at $P_3(x) = x^3 - 3/5x$.

Now that we have our family of polynomials we set about finding the roots. Using Newton's Method, $x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$ makes short work out of an otherwise difficult task. I used Matlab, but

Maple and Mathematica also have root finders. The roots and the respective coefficients are:

n	roots	coefficients
2	.5773502	1
	-.5773502	1
3	.7745966	.5555556
	0	.8888889
	-.7745966	.5555556

We can now use these values to calculate the value of the integral:

$$\int_{-1}^{1} 5x^5 - 7x^3 + x + 4dx = \tag{7}$$

$$.\bar{5}(5(.7745966)^5 - 7(.7745966)^3 + .7745966 + 4) + .\bar{8} * 4 + \tag{8}$$

$$.\bar{5}(5(-.7745966)^5 - 7(-.7745966)^3 - .7745966 + 4) \tag{9}$$

which my computer calculated to be 8.000000. Integrating the left hand side, we arrive at $\frac{5}{6}x^6 - \frac{7}{4}x^4 + \frac{x^2}{2} + 4x|_{-1}^{1} = \frac{43}{12} - \frac{-53}{12} = 8$. How cool is that? We plugged in the roots and weights from a third degree polynomial and got an exact result of a fifth degree polynomial. In other words with a third degree polynomial we were able to calculate the integral of a fifth degree polynomial exactly. So where would you use this? Taylor series comes to mind, suppose you have a nasty integral, which you can not evaluate by hand, you could approximate it with a Taylor series and then use Gaussian Quadrature and get a respectable answer.