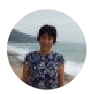
Computational Algorithm for Higher Order Legendre Polynomial and Gaussian Quadrature Method.

CONFERENCE PAPER · JANUARY 2006

Source: DBLP

3 AUTHORS:

Asif Mahmood Mughal
University of Arkansas at Little Rock



Xiu Ye
University of Arkansas at Little Rock

Kamran Iqbal
University of Arkansas at Little Rock

Computational Algorithm for Higher Order Legendre Polynomial and Gaussian Quadrature Method

Asif M. Mughal[1], Xiu Ye[2] and Kamran Iqbal[3]
Dept. of Applied Science[1], Dept. of Mathematics and Statistics[2], Dept. of Systems Engineering[3]
University of Arkansas at Little Rock

Abstract: - **There are many numerical methods adopted to solve mathematical problems. Early researchers focused on the methods to reduce computational costs. In recent years, reduction in computational costs makes many numerical methods available which were not tried for this reason. The use of higher order Legendre polynomials for more than 5-7 orders is usually not common. The efficient and quick numerical methods like Gaussian Quadrature were not adopted for higher orders. In this paper a very simple computational algorithm is adopted for calculations of higher order Legendre polynomials and its use for Gaussian quadrature numerical integration till 44th order.**

I. INTRODUCTION

There are many efficient numerical methods which are not in use much due to computational costs and nature of precise calculations in the computers. In recent years, reduction in technological costs leads to make many solutions available which were only tried and used for very high paid defense and commercial use. Adrien Marie Legendre, a French mathematician (d. 1833) when discovered famous polynomial bear his name now was never aware of that how much it will be used in developing mathematics. This Legendre polynomial is being used by mathematicians and engineers for variety of mathematical and numerical solutions. This use of Legendre polynomial in normal texts is usually only referred till 5-7 order polynomials, because finding Legendre polynomials for higher orders is not only computationally tough but also time consuming. The recent research on Legendre polynomial such as Ref. [1] discusses and explores the use of Legendre series and best leading coefficients of Legendre polynomials for different applications. Ref [2] uses sixth order Legendre polynomial for propagation of guided waves and similar method with higher order polynomials leads to better results. In numerical analysis and methods, Legendre polynomials are used to efficiently calculate numerical integrations by Gaussian quadrature method. This method is very effective in approximating integrals with accuracy and in small time. Ref [3] describes many properties of this method for arbitrary functions and use of Gaussian quadrature for the triangular finite elements. The other researches such as Ref [4-6] published valuable use of this method over period of time, but use of this method for higher order computation requires higher order Legendre polynomials and coefficients of Gaussian quadrature which are usually not available easily in referenced text.

In this paper, first mathematical background is presented followed by computational algorithm and at the end an example is discussed with numerical integrations. Roots of Legendre polynomials and coefficients of Gaussian quadrature are provided in Appendix for 44th order.

II.MATHEMATIAL BACKGROUND

A. *Legendre Polynomials*

The *nth* order Legendre polynomial is generally give by the following equation.

$$P_n(x) = \sum_{m=0}^{M} (-1)^m \left(\frac{n!}{(2n)!}\right) \frac{(2n-2m)!}{m!(n-m)!(n-2m)!} x^{n-2m} \quad (1)$$

Where n is the order of polynomial and M determines the even and odd nature of polynomial by finding which is integer.

$$M = \frac{n}{2} \quad or \quad M = \frac{(n-1)}{2} \quad (2)$$

There are different cases of Legendre polynomials and usually the effect is determined by leading coefficients of x. In original Equation given in Ref [7], leading coefficients of Legendre polynomial is not 1 and scaled by 2^{-n}, but dividing the whole polynomial with leading coefficient with will results in a polynomial given by Eq(1) with leading coefficient 1. The basic property of Legendre polynomials is that these are orthogonal to each other with respect to weight function *w(x) =1* on [-1,1]. The first two polynomials are always same in all cases but the higher order polynomials are created with recursive algorithm and can be normalized accordingly. The set of Legendre polynomial up till any degree n provided an orthogonal basis for different applications. The norm of Legendre polynomials is given by

$$\|P_n(x)\| = \sqrt{\frac{2}{2n+1}} \quad (3)$$

Legendre polynomials till 10th order are given as follows

$$P_0(x) = 1$$
$$P_1(x) = x$$
$$P_2(x) = x^2 - \frac{1}{3}$$
$$P_3(x) = x^3 - \frac{3}{5}x$$
$$P_4(x) = x^4 - \frac{6}{7}x^2 + \frac{3}{35}$$
$$P_5(x) = x^5 - \frac{10}{9}x^3 + \frac{5}{21}x$$
$$P_6(x) = x^6 - \frac{15}{11}x^4 + \frac{5}{11}x^2 - \frac{5}{231}$$
$$P_7(x) = x^7 - \frac{21}{13}x^5 + \frac{105}{143}x^3 - \frac{35}{429}x$$
$$P_8(x) = x^8 - \frac{28}{15}x^6 + \frac{14}{13}x^4 - \frac{28}{143}x^2 + \frac{7}{1287}$$
$$P_9(x) = x^9 - \frac{36}{17}x^7 + \frac{126}{85}x^5 - \frac{84}{221}x^3 + \frac{17}{656}$$
$$P_{10}(x) = x^{10} - \frac{45}{19}x^8 + \frac{630}{323}x^6 - \frac{210}{323}x^4 + \frac{106}{1413}x^2 - \frac{1}{733}$$

The pyramid of Legendre polynomial can be developed at any required degree. These polynomials are generated from code given in sec III. The graphs of Legendre polynomials intercept the x-axis at n times within [-1, 1]. The important thing to note is that the area covered by these polynomials is different and that can be implied to get numerical integration. Fig 1. shows the plot of Legendre polynomials and it can be noted that higher degrees has lesser amplitude.

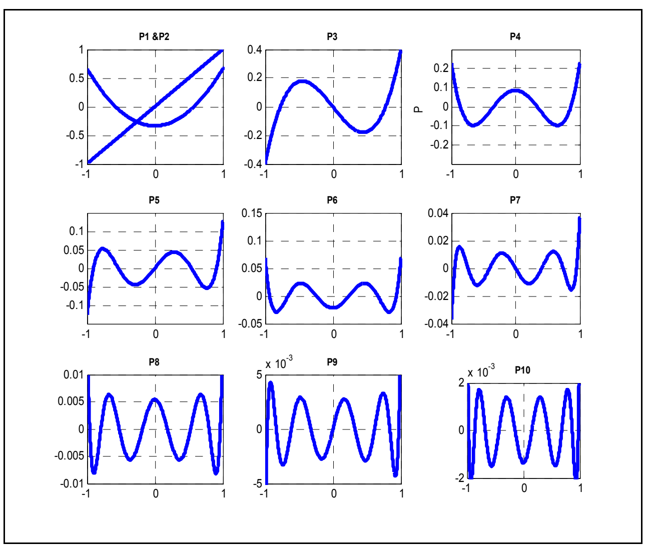


Fig1. Plots of Legendre Polynomials

It is obvious from the figure that increasing order of Legendre polynomials decreases the magnitude of oscillations in the polynomial and this is the property which was exploited by Karlin and Studden to introduce Gaussian quadrature algorithm.

B. Gaussian Quadrature Method

Among many other numerical integration methods, Gaussian quadrature method is used for many integral approximations. The main problem of this method is that it requires computation of Legendre polynomials to get it solution. The algorithm usually follows by the increasing the order of polynomial if error bound doesn't satisfies. Though it is fairly simple algorithm yet it is rarely used in practical applications due to its computational expensiveness. This method works with the roots of Legendre polynomials and finding coefficients for using these roots. The desired function (integrand) is represented in terms of polynomial orthogonal basis. If the integrand is a polynomial then residual error will be zero and Gaussian quadrature is applied as

$$\int_{-1}^{1} P(x) = \sum_{i=1}^{n} c_i P(x_i) \qquad (4)$$

Where the coefficients c_i are given as

$$c_i = \int_{-1}^{1} \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} dx \qquad (5)$$

If the integrand is not a polynomial or not in interval [-1,1] then it can be transformed in to an integral over [-1,1] using change of variables, and in this case error will not be zero.

$$\int_a^b f(x)dx = \int_{-1}^{1} f\left(\frac{(b-a)t + a + b}{2}\right)\frac{(b-a)}{2}dt \qquad (6)$$

The error in this case will converge to zero at higher orders of Legendre polynomials and yield better results. The proof of convergence and details description is given in Ref. [8] pp220-226. In Ref. [9] pp138-139, this method is used for an example to show faster and direct convergence than Simpson's rule or Romberg integration method. However, this is not for all functions and lower order Legendre polynomial with Gaussian quadrature may yield to a non-convergent behavior as shown in example in section IV.

III. COMPUTATIONAL ALGORITHIM

Higher order Legendre polynomials can be obtained by using new computational techniques and also can be applied in different methods such as Gaussian quadrature for numerical integrations. In this paper, a MATLAB code is presented that shows how simply a numerical integration using Gaussian quadrature is possible. The symbolic math toolbox is useful in computing this algorithm. In first coefficients of Legendre polynomial are found using Eq (1) (and also normalize it i.e. leading coefficient is 1). Then zeros are stuffed in coefficients vector to generate roots of polynomial (required by the MATLAB code). The roots are then used to generate coefficients of Gaussian quadrature method as given by Eq (6). Simultaneously Gaussian quadrature integration of a required function is also generated. This method work recursively with even and odd polynomials and get results till the required degree. It is also important note that the computational efficiency of machine on which MATLAB code works also taken into considerations. This code was run on MATLAB v7.5 on Sun Solaris System to get roots and coefficients given in Appendix till 44th order of the polynomial.

A. MATLAB CODE

```
clear all
clc
GQ=0;
soln=0;
F=inline('f(x),'x');
n=1;
MN=44;
c=zeros(MN,MN);
pl=zeros(MN,MN);
for n=1:MN
  if mod(n,2)==0
    a=0;
    M=n/2;
    for m=0:M
      a(m+1)=power(-1,m)*factorial(2*n-
2*m)/(factorial(m)*factorial(n-m)*factorial(n-2*m));
    end
    b=a./a(1);
    j=1;
    i=1;
    rt=zeros(1,2*length(b)-1);
    while j<=length(rt)
      rt(j)=b(i);
      j=j+2;
      i=i+1;
    end
    p=roots(rt);
    for u=1:length(p)
    pl(u,n)=p(u);
  end
  syms x;
  i=1;
  j=1;
  G=1;
  H=1;
  for i=1:n
    for j=1:n
      if j==i
        l=0;%disp(i);
      else
        G=G*(x-p(j))/(p(i)-p(j));
      end
    end
    c(i,n)=double(int(G,x,-1,1));
    H=G;
    G=1;
    end
    sum=0;
    for i=1:n
      sum=sum+c(i,n)*F(p(i));
    end
    GQ(n)=sum;
    else
      a=0;
      M=(n-1)/2;
      for m=0:M
        a(m+1)=power(-1,m)*factorial(2*n-
2*m)/(factorial(m)*factorial(n-m)*factorial(n-2*m));
      end
      b=a./a(1);
      j=1;
      i=1;
      rt=zeros(1,2*length(b));
      while j<=length(rt)
        rt(j)=b(i);
        j=j+2;
        i=i+1;
      end
      p=roots(rt);
      for u=1:length(p)
        pl(u,n)=p(u);
      end
      syms x;
      i=1;
      j=1;
      c=0;
      G=1;
      H=1;
      for i=1:n
        for j=1:n
          if j==i
            l=0;%disp(i);
          else
            G=G*(x-p(j))/(p(i)-p(j));
          end
        end
        c(i,j)=double(int(G,x,-1,1));
        H=G;
        G=1;
      end
    sum=0;
      for i=1:n
        sum=sum+c(i,j)*F(p(i));
      end
      GQ(n)=sum;
    end
  soln=GQ(n)
  end

%END PROGRAM
```

This code generates matrices for roots of Legendre polynomials and coefficients c_i for Gaussians quadrature method for 1 to *MN*, (*nth*) order. Solution of Gaussian quadrature *GQ* is not required in this code, it can be done separately and easily, but it is provided in this to see the trend of convergence with respect to order of polynomials. The roots of any m order polynomial can be store in a vector form in one variable e.g. *p* and a simple MATLAB command *rats(poly(p))* will give a *mth* order polynomial in fractions.

IV. EXAMPLE

The code given in previous section can be computed for any function of ***x*** provided that it must be generated according to Eq (6), and integration limits are appropriately transformed with in [-1,1] bound. There are many functions for which Gaussian quadrature converges in a low order and proves the better results than Simpson's rule or Romberg integration. But there are many for which if roots of Legendre polynomials and coefficients of Gaussian quadrature are available, it can compute the results with less computational delays. One such example is the function given in Eq (7), this seems to be a very easy problem and the exact solution of this can be obtained very easily.

$$I = \int_{-10}^{10} \frac{1}{x^2+1} dx \tag{7}$$

The exact solution of this integral is given as

$$I = \tan^{-1}(x)\Big|_{-10}^{10} = 2.9422553486075 \tag{8}$$

In this case we are considering an absolute error bound less than 10^{-3}. The integral of Eq (7) can be transformed according to Eq(6) and it will be given as

$$I = \int_{-1}^{1} \frac{10}{100t^2+1} dt \tag{9}$$

Transforming the integrand *f(x)* in Eq(7) and Eq(9) actually doesn't change the functional properties as shown in Fig 2.

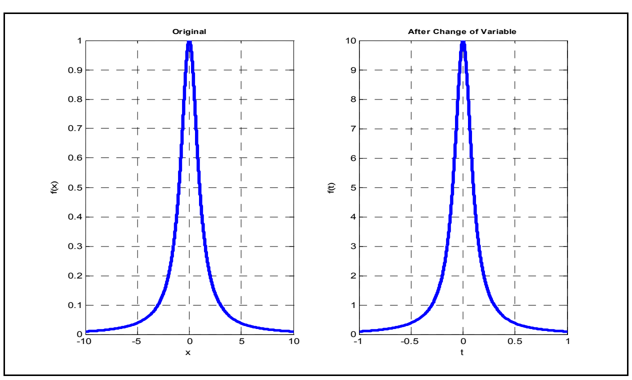


Fig 2. Integrand before and after change of variable

It can be seen the function from -10 to 10 remains the same within -1 to 1 after change of variable and in original case the maximum value is 1 and after change of variable the maximum value is 10. Such functions to numerical integrate with low order and using Simpson's Rule and Romberg Integration the values are given in Table 1. These methods are not discussed here and their discussion is available in Ref [8], pp196-213. Also to note that *n* in case of Romberg integration method, it is not simple as it computes *n x n* matrices. The convergence of this problem with respect to order of Gaussian quadrature is shown in Table 2. It is also noted that in case of available roots and coefficients, the calculation method is simpler than Simpson's rule and Romberg Integration method. In the above code, in an *inline* function, left hand side of Eq (7) is used instead of *f(x)* before running the code for this example. At *n = 44*, the absolute error is *0.00107540847479* with respect to exact solution, better than the similar order of Simpson's rule or Romberg integration method.

TABLE 1
Integration with Simpson's Rule and Romberg Method

Simpson's Rule	n=54	2.94268932
Simpson's Rule	n=94	2.94225615
Romberg Method	n=8	2.94224394
Romberg Method	n=9	2.94225529

TABLE 2
Gaussian Quadrature Method

n	GQ	n	GQ
1	20.0000000000	23	3.0003208752
2	0.5825242718	24	2.8954837430
3	9.0710382514	25	2.9810887634
4	1.1311235558	26	2.9108031188
5	6.0650397187	27	2.9682523996
6	1.5983311330	28	2.9211220756
7	4.7450439611	29	2.9596705404
8	1.9717903653	30	2.9280633942
9	4.0480178435	31	2.9539274861
10	2.2560912728	32	2.9327282367
11	3.6436901244	33	2.9500794148
12	2.4646698608	34	2.9358612855
13	3.3962276447	35	2.9474934816
14	2.6136004512	36	2.9379640415
15	3.2397622941	37	2.9457604012
16	2.7178974820	38	2.9393869306
17	3.1387863801	39	2.9446632928
18	2.7899495835	40	2.9403162862
19	3.0727571047	41	2.9433495174
20	2.8392588486	42	2.9409256939
21	3.0292072635	43	2.9460237223
22	2.8727866309	44	2.9411799401

V. CONCLUSION

The roots or Legendre polynomial can be calculated easily with the use of simple MATLAB code and higher order polynomials are available to use in different applications. The higher order roots are successfully used to apply in Gaussian quadrature integration method for a problem which seems to be unsolvable by this method in lower orders. Any order polynomial can be generated for other applications using this code.

APPENDIX -1

TABLE 3
Calculated roots of Legendre Polynomial and Gaussian quadrature coefficients for 44th order.

c_{44}	r_{44}
0.00343377843623	0.99832148372556
0.00913091418328	0.99296411747259
0.01390799768469	0.98001813909306
0.01714770681113	0.96642679367651
0.02580754140856	0.94308861061881
0.02455534105973	0.91950051428355
0.03571332227180	0.88809834561065
0.03363032689591	0.85414813463275
0.04350847660426	0.81510233478855
0.04264937566655	0.77224708426622
0.05040561446090	0.72555985884403
0.05053607275321	0.67516102361618
0.05660297588543	0.62149342571538
0.05715419489273	0.56466428627135
0.00337711368703	-0.99826447411502
0.00916045474585	-0.99309897258724
0.01409885555726	-0.97981639877940
0.01680513270491	-0.96662791501718
0.02628992610415	-0.94294379644525
0.02400036660825	-0.91960729951730
0.03623923659374	-0.88803206256316
0.03310897337413	-0.85419212085749
0.04399288545043	-0.81507542259352
0.04219142738853	-0.77226216959427
0.05083169543308	-0.72555273320246
0.05013773563296	-0.67516365527984
0.05697749144971	-0.62149279457189
0.05679854478119	-0.56466428478834
0.06186286933960	0.50505768217107
0.06220427750410	-0.50505777065882
0.06251064930027	0.44291915380475
0.06218009024039	-0.44291910544655
0.06596774051490	0.37857958073018
0.06629005155839	-0.37857959604763
0.06657699950240	0.31235243393281
0.06626101534920	-0.31235243084395
0.06878927760645	0.24456945908297
0.06910049865691	-0.24456945944619
0.06929793698758	0.17556801477940
0.06899013893173	-0.17556801476066
0.07025569560419	0.10569190170671
0.07056127731285	-0.10569190170690
0.07063124770057	0.03528923696413
0.07032675536488	-0.03528923696413

REFERENCES

[1] Gavin Brown, Stamatis Koumandos, and Kunyang Wang. "On the positivity of some basic legendre polynomial sums" Journal of London Mathematical Society, 59, pp939-954, Cambridge University Press, 1999.

[2] .Jean E. Lefebvre, Victor Zhang, Joseph Gazalet, Tadeusz Gryba, and Veronique Sadaune, "Acoustic Wave Propagation in Continuous Functionally Graded Plates: An Extension of the Legendre Polynomial Approach", IEEE Transactions On Ultrasonics, Ferroelectrics, And Frequency Control, Vol. 48, No. 5, September 2001

[3] D. A. Dunavant. High degree efficient symmetrical Gaussian quadrature rules for the triangle". International Journal of Numerical Methods in Engineering", Vol. 21, No. 6, pp1129-1148, 1985.

[4] J. Ma, V. Rokhlin, S. Wandzura "Generalized Gaussian Quadrature Rules for Systems of Arbitrary Functions". SIAM Journal on Numerical Analysis, Vol. 33 No.3, pp. 971-996. 1996.

[5] Dirk P. Laurie, "Calculation Of Gauss-Kronrod Quadrature Rules", Mathematics Of Computation, Vol. 66, No. 219, pp 133-1145, July 1997.

[6] Carlos F. Borges, "On a class of Gauss-like quadrature rules", Numerische Mathematik, Volume 67, No. 3, pp 271-288, April 1994

[7] Erwin Kreyszig, "Advanced Engineeing Mathematics", 7th Ed, J. Weily, 1993.

[8] J.D. Faires, R. Burden, Numerical Analysis, 7th Ed., Brooks/Cole Publishing, 2001.

[9] J.D. Faires, R. Burden, Numerical Methods, 2nd Ed., Brooks/Cole Publishing, 1998.