Faculty of Mathematics and Information Science
WARSAW UNIVERSITY OF TECHNOLOGY

Numerical Methods 2
Gauss–Legendre quadrature

Gauss–Legendre quadrature

Piotr Możeluk 313122

November 10, 2023

1 Title of a task

14. Numerical calculation of the integral $\int_a^b f(x)dx$. Use the n-point (n = 3,5,7,...,33) Gauss-Legendre rule.

2 Description of Gauss-Legendre rule

The Gauss-Legendre rule is a form of Gaussian quadrature for approximating the definite integral of a function over the interval [-1, 1]. It uses n sample points, called nodes, that are the roots of the n-th Legendre polynomial, and n corresponding weights that are computed from a formula involving the derivative of the Legendre polynomial. The Gauss-Legendre rule can integrate polynomials of degree up to 2n - 1 exactly and is more accurate than other numerical integration methods such as the trapezoidal rule or Simpson's rule for smooth integrands. The Gauss-Legendre rule can be generalized to other intervals. The rule takes a form:

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \tag{1}$$

where:

- n is the number of sample points
- wi_i are quadrature weights
- x_i are the roots of the nth Legendre polynomial

3 Gauss-Legendre rule for all real intervals

For integrating over a general real interval [a,b], a change of interval must be applied to convert the problem to one of integrating over [-1,1] before applying the Gaussian-Legendre quadrature rule [**1**]. This change of interval can be done in the following way:

$$\int_a^b f(x)\,dx = \int_{-1}^{1} f\left(\frac{b-a}{2}\xi + \frac{a+b}{2}\right)\frac{dx}{d\xi}d\xi \tag{2}$$

with : $\frac{dx}{d\xi} = \frac{b-a}{2}$
Applying the n-point Gaussian-Legendre quadrature rule then results in the following approximation:

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}\sum_{i=1}^{n} w_i f\left(\frac{b-a}{2}\xi_i + \frac{a+b}{2}\right). \tag{3}$$

where:

- a and b are the endpoints of the original interval of integration
- n is the number of sample points
- w_i are the weights associated with each node ξ_i
- ξ_i are the nodes which are the roots of the i-th Legendre polynomial

4 Finding the weights and nodes

Weight can be easily calculated using the following formula from Abramowitz's and Stegun's Handbook of Mathematical functions[**2**]:

$$w_i = \frac{2}{(1 - x_i^2)\left[P_n'(x_i)\right]^2} \tag{4}$$

where:

- $P_n(x)$ denotes Legendre polynomials With the n-th degree and normalized to satisfy condition $P_n(1) = 1$
- w_i are the weights associated with each node x_i
- x_i are the nodes or sample points, which are the roots of the n-th Legendre polynomial

Finding the roots of the Legendre polynomial is a little harder because there isn't any formula to find those. We will need to use Newton's methods to approximate the roots. So the formula after some iterations will look like this:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{5}$$

Now the problem is to find an initial guess for Newton's method. In the paper "Sull'approssimazione asintotica degli zeri dei polinomi sferici ed ultrasferici" Luigi Gatteschi[**3**] proposes an initial guess which is similar to Chebyshev root which can be obtained from the formula in the following form:

$$\xi_k \approx (1 - \frac{1}{n^2}(1 - \frac{1}{n}))x_k + \varepsilon \tag{6}$$

where:

$$x_k = cos(\frac{4k-1}{4n+2}\pi)$$

and

$$|\varepsilon| < \frac{5}{2n^4}$$

- ξ_k is the k-th node of the n-point Gauss-Legendre rule, which is the root of the n-th Legendre polynomial
- x_k which is an approximation of the Gauss-Legendre node
- n is the number of sample points used in the Gauss-Legendre rule
- k is the index of the node, ranging from 1 to n

5 Legendre polynomials

In formulas (4) and (5) we have the Legendre polynomial and the derivative of this polynomial however it hasn't been defined yet. The Legradre polynomial can be defined as the three-term recurrence relation known as Bonnet's recursion formula given by:

$$P_0(x) \equiv 1, \quad P_1(x) = x, \quad P_k(X) = \frac{2k-1}{k} x P_{k-1}(X) - \frac{k-1}{k} P_{k-2}(x) \quad (k > 1)$$

and also:

$$P_n'(x) = \frac{n}{x^2-1}(xP_n(x) - P_{n-1}(x)$$

Now we have every formula to implement the algorithm for computing integrals using the Gauss–Legendre rule

6 Observation

For odd values of n, the n-point Gauss-Legendre rule has a node at x=0, and the other nodes are symmetrically distributed around zero. For example, the 3-point rule has nodes at $x = 0, \pm\sqrt{\frac{3}{5}}$ and equivalent weights at $w = \frac{8}{9}, \frac{5}{9}, \frac{5}{9}$. This relation between nodes and weight reduces the required computation by $\lfloor \frac{n}{2} \rfloor$.

7 Testing

The crucial step of the algorithm is to find the correct nodes and weights for the Gauss–Legendre rule on the interval [-1,1]. Figure 1 shows the average errors in the computation for each n. The exact values of the coefficients and nodes are taken from the file available at this link [https://github.com/ananasek727/Gauss_Legendre_quadrature]. We are confident that our algorithm computes the integral correctly, as it has a very low error rate.

8 Bibliography

References

[1] Changing the limits of integrals: https://en.wikipedia.org/wiki/Gaussian_quadrature#Change_of_interval

[2] Milton Abramowitz, Irene A. Stegun, Robert H. Romer; Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, American Journal of Physics, 56(10), 958-959 (1988).

[3] Gatteschi, Luigi. "Sull'approssimazione asintotica degli zeri dei polinomi sferici ed ultrasferici." Bollettino dell'Unione Matematica Italiana 5.3-4 (1950): 305-313.

N	Coefficient error	Node error
3	1.48029736616688e-16	0
5	1.11022302462516e-17	0
7	2.37904933848248e-17	0
9	1.60365548001411e-16	2.46716227694479e-17
11	9.20980463609505e-17	0
13	5.23085848140699e-17	8.5401771125012e-18
15	7.54026470891252e-17	0
17	3.18372779120449e-17	0
19	5.84327907697451e-17	0
21	6.17891980966977e-17	0
23	7.04448576766234e-17	4.82705662880503e-18
25	5.96744875736022e-17	1.33226762955019e-17
27	7.81268054365851e-17	0
29	5.77842371652964e-17	7.65671051465625e-18
31	4.57183574152597e-17	0
33	4.42091649483786e-17	3.36431219583381e-18
35	5.12486878331434e-17	1.11022302462516e-17
37	4.74001468706095e-17	0
39	2.81558964177774e-17	5.69345140833414e-18
41	4.2521880325773e-17	5.41572207134223e-18
43	2.9651668717278e-17	5.16382802151236e-18
45	4.62978421032921e-17	4.93432455388958e-18
47	4.02861845538018e-17	0
49	3.83940328509561e-17	5.66440318686304e-18
51	3.35890084809235e-17	7.8912911064043e-18
53	3.93749498415113e-17	7.33166148337368e-18
55	2.68093628105507e-17	5.80343853781332e-18
57	3.09054682430605e-17	9.49532850008358e-18
59	3.28421376723067e-17	3.52825113757995e-18
61	3.63012215915474e-17	1.18302453443664e-17
63	3.32282151529367e-17	7.26931742314091e-18
65	2.97171475461565e-17	2.98906198937542e-18
67	3.72900818847104e-17	2.07131161310664e-19
69	3.33368598426847e-17	3.41916511207023e-18
71	3.40653268504319e-17	4.49562140253144e-18
73	3.13735434131285e-17	8.55480070344727e-18
75	2.15394831600454e-17	3.14563190310461e-18
77	2.93551258337374e-17	0
79	3.11591469925454e-17	4.21603680237401e-18
81	2.52230935040949e-17	8.22387425648264e-18
83	2.85184359394018e-17	8.86171992547188e-18
85	3.45822227063847e-17	1.01226216951117e-17
87	2.57242543039607e-17	4.78544407166016e-19
89	3.00604188856767e-17	8.73209120491696e-18
91	4.0696898689846e-17	2.59255376629501e-18
93	2.76926219410975e-17	2.68602344667377e-18
95	3.01750583584387e-17	4.82070523850397e-18
97	3.14373920652859e-17	2.57525959320268e-18
99	1.934479512604444e-17	2.80359349652817e-18
101	3.40718583709801e-17	2.4045671944233e-18

Figure 1: Mean errors