Gaussian quadrature

In numerical analysis, a **quadrature rule** is an approximation of the definite integral of a function, usually stated as a weighted sum of function values at specified points within the domain of integration. (See numerical integration for more on quadrature rules.) An n-point **Gaussian quadrature rule**, named after Carl Friedrich Gauss, is a quadrature rule constructed to yield an exact result for polynomials of degree $2n-1$ or less by a suitable choice of the points x_i and weights w_i for $i = 1,...,n$. The domain of integration for such a rule is conventionally taken as [−1, 1], so the rule is stated as

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

Gaussian quadrature as above will only produce accurate results if the function $f(x)$ is well approximated by a polynomial function within the range [-1,1]. The method is not, for example, suitable for functions with singularities. However, if the integrated function can be written as $f(x) = W(x)g(x)$, where $g(x)$ is approximately polynomial, and $W(x)$ is known, then there are alternative weights w_i' such that

$$\int_{-1}^{1} f(x)\,dx = \int_{-1}^{1} W(x)g(x)\,dx \approx \sum_{i=1}^{n} w_i' g(x_i).$$

Common weighting functions include $W(x) = (1-x^2)^{-1/2}$ (Chebyshev–Gauss) and $W(x) = e^{-x^2}$ (Gauss–Hermite).

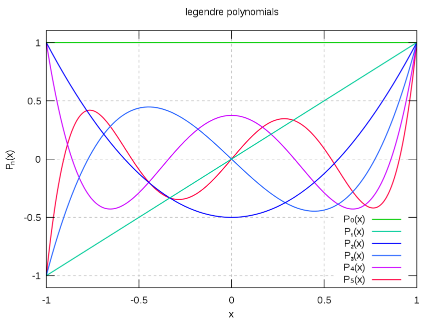
It can be shown (see Press, et al., or Stoer and Bulirsch) that the evaluation points are just the roots of a polynomial belonging to a class of orthogonal polynomials.

Gauss–Legendre quadrature

For the simplest integration problem stated above, i.e. with $W(x) = 1$, the associated polynomials are Legendre polynomials, $P_n(x)$, and the method is usually known as Gauss–Legendre quadrature. With the n^{th} polynomial normalized to give $P_n(1) = 1$, the i^{th} Gauss node, x_i, is the i^{th} root of P_n; its weight is given by (Abramowitz & Stegun 1972, p. 887)

$$w_i = \frac{2}{(1-x_i^2)\,[P_n'(x_i)]^2}$$

Some low-order rules for solving the integration problem are listed below.



Graphs of Legendre polynomials (up to n = 5)

Number of points, n	Points, x_i	Weights, w_i
1	0	2
2	$\pm 1/\sqrt{3}$	1
3	0	$^8/_9$
	$\pm\sqrt{3/5}$	$^5/_9$
4	$\pm\sqrt{\left(3-2\sqrt{6/5}\right)/7}$	$\frac{18+\sqrt{30}}{36}$
	$\pm\sqrt{\left(3+2\sqrt{6/5}\right)/7}$	$\frac{18-\sqrt{30}}{36}$

5	0	128/225
	$\pm\frac{1}{3}\sqrt{5-2\sqrt{10/7}}$	$\frac{322+13\sqrt{70}}{900}$
	$\pm\frac{1}{3}\sqrt{5+2\sqrt{10/7}}$	$\frac{322-13\sqrt{70}}{900}$

Change of interval

An integral over [a, b] must be changed into an integral over [−1, 1] before applying the Gaussian quadrature rule. This change of interval can be done in the following way:

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

After applying the Gaussian quadrature rule, the following approximation is:

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

Other forms

The integration problem can be expressed in a slightly more general way by introducing a positive weight function ω into the integrand, and allowing an interval other than [−1, 1]. That is, the problem is to calculate

$$\int_a^b \omega(x)\,f(x)\,dx$$

for some choices of a, b, and ω. For a = −1, b = 1, and ω(x) = 1, the problem is the same as that considered above. Other choices lead to other integration rules. Some of these are tabulated below. Equation numbers are given for Abramowitz and Stegun (A & S).

Interval	ω(x)	Orthogonal polynomials	A & S	For more information, see ...
[−1, 1]	1	Legendre polynomials	25.4.29	Section *Gauss–Legendre quadrature*, above
(−1, 1)	$(1-x)^\alpha(1+x)^\beta,\quad \alpha,\beta>-1$	Jacobi polynomials	25.4.33 ($\beta=0$)	Gauss–Jacobi quadrature
(−1, 1)	$\frac{1}{\sqrt{1-x^2}}$	Chebyshev polynomials (first kind)	25.4.38	Chebyshev–Gauss quadrature
[−1, 1]	$\sqrt{1-x^2}$	Chebyshev polynomials (second kind)	25.4.40	Chebyshev–Gauss quadrature
[0, ∞)	e^{-x}	Laguerre polynomials	25.4.45	Gauss–Laguerre quadrature
(−∞, ∞)	e^{-x^2}	Hermite polynomials	25.4.46	Gauss–Hermite quadrature

Fundamental theorem

Let p_n be a nontrivial polynomial of degree n such that

$$\int_a^b \omega(x)\, x^k p_n(x)\, dx = 0, \quad \text{for all } k = 0, 1, \ldots, n-1.$$

If we pick the n nodes x_i to be the zeros of p_n , then there exist n weights w_i which make the Gauss-quadrature computed integral exact for all polynomials $h(x)$ of degree $2n - 1$ or less. Furthermore, all these nodes x_i will lie in the open interval (a, b) (Stoer & Bulirsch 2002, pp. 172–175).

The polynomial p_n is said to be an orthogonal polynomial of degree n associated to the weight function $\omega(x)$. It is unique up to a constant normalization factor. The idea underlying the proof is that, because of its sufficiently low degree, $h(x)$ can be divided by $p_n(x)$ to produce a quotient $q(x)$ of degree strictly lower than n, and a remainder $r(x)$ of still lower degree, so that both will be orthogonal to $p_n(x)$, by the defining property of $p_n(x)$. Thus

$$\int_a^b \omega(x)\, h(x)\, dx = \int_a^b \omega(x)\, r(x)\, dx.$$

Because of the choice of nodes x_i, the corresponding relation

$$\sum_{i=1}^n w_i h(x_i) = \sum_{i=1}^n w_i r(x_i)$$

holds also. The exactness of the computed integral for $h(x)$ then follows from corresponding exactness for polynomials of degree only n or less (as is $r(x)$).

General formula for the weights

The weights can be expressed as

$$w_i = \frac{a_n}{a_{n-1}} \frac{\int_a^b \omega(x) p_{n-1}(x)^2 dx}{p'_n(x_i) p_{n-1}(x_i)} \quad (1)$$

where a_k is the coefficient of x^k in $p_k(x)$. To prove this, note that using Lagrange interpolation one can express $r(x)$ in terms of $r(x_i)$ as

$$r(x) = \sum_{i=1}^n r(x_i) \prod_{\substack{1 \le j \le n \\ j \ne i}} \frac{x - x_j}{x_i - x_j}$$

because r(x) has degree less than n and is thus fixed by the values it attains at n different points. Multiplying both sides by $\omega(x)$ and integrating from a to b yields

$$\int_a^b \omega(x) r(x) dx = \sum_{i=1}^n r(x_i) \int_a^b \omega(x) \prod_{\substack{1 \le j \le n \\ j \ne i}} \frac{x - x_j}{x_i - x_j} dx$$

The weights w_i are thus given by

$$w_i = \int_a^b \omega(x) \prod_{\substack{1 \le j \le n \\ j \ne i}} \frac{x - x_j}{x_i - x_j} dx$$

This integral expression for w_i can be expressed in terms of the orthogonal polynomials $p_n(x)$ and $p_{n+1}(x)$ as follows.

We can write

$$\prod_{\substack{1 \le j \le n \\ j \ne i}} (x - x_j) = \frac{\prod_{1 \le j \le n} (x - x_j)}{x - x_i} = \frac{p_n(x)}{a_n (x - x_i)}$$

where a_n is the coefficient of x^n in $p_n(x)$. Taking the limit of x to x_i yields using L'Hopital's rule

$$\prod_{\substack{1\le j\le n\\ j\neq i}} (x_i - x_j) = \frac{p'_n(x_i)}{a_n}$$

We can thus write the integral expression for the weights as

$$w_i = \frac{1}{p'_n(x_i)} \int_a^b \omega(x) \frac{p_n(x)}{x - x_i} dx \quad (2)$$

In the integrand, writing

$$\frac{1}{x - x_i} = \frac{1 - \left(\frac{x}{x_i}\right)^k}{x - x_i} + \left(\frac{x}{x_i}\right)^k \frac{1}{x - x_i}$$

yields

$$\int_a^b \omega(x) \frac{x^k p_n(x)}{x - x_i} dx = x_i^k \int_a^b \omega(x) \frac{p_n(x)}{x - x_i} dx$$

provided $k \le n$, because $\frac{1 - \left(\frac{x}{x_i}\right)^k}{x - x_i}$ is a polynomial of degree k-1 which is then orthogonal to $p_n(x)$. So, if q(x) is a polynomial of at most nth degree we have

$$\int_a^b \omega(x) \frac{p_n(x)}{x - x_i} dx = \frac{1}{q(x_i)} \int_a^b \omega(x) \frac{q(x) p_n(x)}{x - x_i} dx$$

We can evaluate the integral on the right hand side for $q(x) = p_{n-1}(x)$ as follows. Because $\frac{p_n(x)}{x - x_i}$ is a polynomial of degree n-1, we have

$$\frac{p_n(x)}{x - x_i} = a_n x^{n-1} + s(x)$$

where s(x) is a polynomial of degree n-2. Since s(x) is orthogonal to $p_{n-1}(x)$ we have

$$\int_a^b \omega(x) \frac{p_n(x)}{x - x_i} dx = \frac{a_n}{p_{n-1}(x_i)} \int_a^b \omega(x) p_{n-1}(x) x^{n-1} dx$$

We can then write

$$x^{n-1} = \left(x^{n-1} - \frac{p_{n-1}(x)}{a_{n-1}}\right) + \frac{p_{n-1}(x)}{a_{n-1}}$$

The term in the brackets is a polynomial of degree n-2, which is therefore orthogonal to $p_{n-1}(x)$. The integral can thus be written as

$$\int_a^b \omega(x) \frac{p_n(x)}{x - x_i} dx = \frac{a_n}{a_{n-1} p_{n-1}(x_i)} \int_a^b \omega(x) p_{n-1}(x)^2 dx$$

According to Eq. (2), the weights are obtained by dividing this by $p'_n(x_i)$ and that yields the expression in Eq. (1).

Proof that the weights are positive

Consider the following polynomial of degree 2n-2

$$f(x) = \prod_{\substack{1 \le j \le n \\ j \ne i}} (x - x_j)^2$$

where as above the x_j are the roots of the polynomial $p_n(x)$. Since the degree of f(x) is less than 2n-1, the Gaussian quadrature formula involving the weights and nodes obtained from $p_n(x)$ applies. Since $f(x_j) = 0$ for j not equal to i, we have

$$\int_a^b \omega(x) f(x) = \sum_{j=1}^{N} w_j f(x_j) = w_i f(x_i)$$

Since both $\omega(x)$ and f(x) are non-negative functions, it follows that $w_i > 0$.

Computation of Gaussian quadrature rules

For computing the nodes x_i and weights w_i of Gaussian quadrature rules, the fundamental tool is the three-term recurrence relation satisfied by the set of orthogonal polynomials associated to the corresponding weight function. For *n* points, these nodes and weights can be computed in $O(n^2)$ operations by the following algorithm.

If, for instance, p_n is the monic orthogonal polynomial of degree *n* (the orthogonal polynomial of degree *n* with the highest degree coefficient equal to one), one can show that such orthogonal polynomials are related through the recurrence relation

$$p_{n+1}(x) + (B_n - x)p_n(x) + A_n p_{n-1}(x) = 0, \qquad n = 1, 2, \ldots$$

From this, nodes and weights can be computed from the eigenvalues and eigenvectors of an associated linear algebra problem. This is usually named as the Golub–Welsch algorithm (Gil, Segura & Temme 2007).

The starting idea comes from the observation that, if x_i is a root of the orthogonal polynomial p_n then, using the previous recurrence formula for $k = 0, 1, \ldots, n-1$ and because $p_n(x_j) = 0$, we have

$$J\tilde{P} = x_j \tilde{P}$$

where $\tilde{P} = [p_0(x_j), p_1(x_j), ..., p_{n-1}(x_j)]^T$

and J is the so-called Jacobi matrix:

$$\mathbf{J} = \begin{pmatrix} B_0 & 1 & 0 & \ldots & \ldots & \ldots \\ A_1 & B_1 & 1 & 0 & \ldots & \ldots \\ 0 & A_2 & B_2 & 1 & 0 & \ldots \\ 0 & \ldots & \ldots & \ldots & \ldots & 0 \\ \ldots & \ldots & 0 & A_{n-2} & B_{n-2} & 1 \\ \ldots & \ldots & \ldots & 0 & A_{n-1} & B_{n-1} \end{pmatrix}.$$

The nodes of gaussian quadrature can therefore be computed as the eigenvalues of a tridiagonal matrix.

For computing the weights and nodes, it is preferable to consider the symmetric tridiagonal matrix $\mathcal{J}$ with elements $\mathcal{J}_{i,i} = J_{i,i} = B_{i-1},\ i = 1, \ldots, n$ and $\mathcal{J}_{i-1,i} = \mathcal{J}_{i,i-1} = \sqrt{J_{i,i-1} J_{i-1,i}} = \sqrt{A_{i-1}},\ i = 2, \ldots, n.$ **J** and $\mathcal{J}$ are similar matrices and therefore have the same eigenvalues (the nodes). The weights can be computed from the corresponding eigenvectors: If $\phi^{(j)}$ is a normalized eigenvector (i.e., an eigenvector with euclidean norm equal to one) associated to the eigenvalue x_j, the corresponding weight can be computed from the first component of this eigenvector, namely:

$$w_j = \mu_0 \left(\phi_1^{(j)}\right)^2$$

where μ_0 is the integral of the weight function

$$\mu_0 = \int_a^b w(x)dx.$$

See, for instance, (Gil, Segura & Temme 2007) for further details.

There are alternative methods for obtaining the same weights and nodes in $O(n)$ operations using the Prüfer Transform.

Error estimates

The error of a Gaussian quadrature rule can be stated as follows (Stoer & Bulirsch 2002, Thm 3.6.24). For an integrand which has $2n$ continuous derivatives,

$$\int_a^b \omega(x)\, f(x)\, dx - \sum_{i=1}^n w_i\, f(x_i) = \frac{f^{(2n)}(\xi)}{(2n)!}\,(p_n, p_n)$$

for some ξ in (a, b), where p_n is the orthogonal polynomial of degree n and where

$$(f, g) = \int_a^b \omega(x) f(x) g(x)\, dx.$$

In the important special case of $\omega(x) = 1$, we have the error estimate (Kahaner, Moler & Nash 1989, §5.2)

$$\frac{(b-a)^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3} f^{(2n)}(\xi), \qquad a < \xi < b.$$

Stoer and Bulirsch remark that this error estimate is inconvenient in practice, since it may be difficult to estimate the order $2n$ derivative, and furthermore the actual error may be much less than a bound established by the derivative. Another approach is to use two Gaussian quadrature rules of different orders, and to estimate the error as the difference between the two results. For this purpose, Gauss–Kronrod quadrature rules can be useful.

Important consequence of the above equation is that Gaussian quadrature of order n is accurate for all polynomials up to degree $2n$–1.

Gauss–Kronrod rules

If the interval $[a, b]$ is subdivided, the Gauss evaluation points of the new subintervals never coincide with the previous evaluation points (except at zero for odd numbers), and thus the integrand must be evaluated at every point. *Gauss–Kronrod rules* are extensions of Gauss quadrature rules generated by adding $n + 1$ points to an n-point rule in such a way that the resulting rule is of order $3n + 1$. This allows for computing higher-order estimates while re-using the function values of a lower-order estimate. The difference between a Gauss quadrature rule and its Kronrod extension are often used as an estimate of the approximation error.

Gauss–Lobatto rules

Also known as **Lobatto quadrature** (Abramowitz & Stegun 1972, p. 888), named after Dutch mathematician Rehuel Lobatto.

It is similar to Gaussian quadrature with the following differences:

1. The integration points include the end points of the integration interval.
2. It is accurate for polynomials up to degree $2n$–3, where n is the number of integration points.

Lobatto quadrature of function $f(x)$ on interval [−1, +1]:

$$\int_{-1}^1 f(x)\, dx = \frac{2}{n(n-1)}[f(1) + f(-1)] + \sum_{i=2}^{n-1} w_i f(x_i) + R_n.$$

Abscissas: x_i is the $(i-1)^{\text{st}}$ zero of $P'_{n-1}(x)$.

Weights:

$$w_i = \frac{2}{n(n-1)[P_{n-1}(x_i)]^2} \quad (x_i \neq \pm 1).$$

Remainder: $R_n = \frac{-n(n-1)^3 2^{2n-1}[(n-2)!]^4}{(2n-1)[(2n-2)!]^3} f^{(2n-2)}(\xi), \quad (-1 < \xi < 1)$

Some of the weights are:

Number of points, *n*	**Points, x_i**	**Weights, w_i**
3	0	$4/3$
	± 1	$1/3$
4	$\pm\sqrt{\frac{1}{5}}$	$5/6$
	± 1	$1/6$
5	0	$32/45$
	$\pm\sqrt{\frac{3}{7}}$	$49/90$
	± 1	$1/10$

References

- Abramowitz, Milton; Stegun, Irene A., eds. (1972), "§25.4, Integration", *Handbook of Mathematical Functions (with Formulas, Graphs, and Mathematical Tables)*, Dover, ISBN 978-0-486-61272-0
- Gil, Amparo; Segura, Javier; Temme, Nico M. (2007), "§5.3: Gauss quadrature", *Numerical Methods for Special Functions*, SIAM, ISBN 978-0-89871-634-4
- Golub, Gene H.; Welsch, John H. (1969), "Calculation of Gauss Quadrature Rules", *Mathematics of Computation* **23** (106): 221–230, doi:10.1090/S0025-5718-69-99647-1, JSTOR 2004418
- Kahaner, David; Moler, Cleve; Nash, Stephen (1989), *Numerical Methods and Software*, Prentice-Hall, ISBN 978-0-13-627258-8
- Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007), "Section 4.6. Gaussian Quadratures and Orthogonal Polynomials" [1], *Numerical Recipes: The Art of Scientific Computing* (3rd ed.), New York: Cambridge University Press, ISBN 978-0-521-88068-8
- Stoer, Josef; Bulirsch, Roland (2002), *Introduction to Numerical Analysis* (3rd ed.), Springer, ISBN 978-0-387-95452-3.
- Temme, Nico M. (2010), "§3.5(v): Gauss Quadrature" [2], in Olver, Frank W. J.; Lozier, Daniel M.; Boisvert, Ronald F. et al., *NIST Handbook of Mathematical Functions*, Cambridge University Press, ISBN 978-0521192255, MR2723248

External links

- ALGLIB [3] contains a collection of algorithms for numerical integration (in C# / C++ / Delphi / Visual Basic / etc.)
- GNU Scientific Library [4] - includes C version of QUADPACK algorithms (see also GNU Scientific Library)
- From Lobatto Quadrature to the Euler constant e [5]
- Gaussian Quadrature Rule of Integration - Notes, PPT, Matlab, Mathematica, Maple, Mathcad [6] at *Holistic Numerical Methods Institute*
- Legendre-Gauss Quadrature at MathWorld [7]
- Gaussian Quadrature [8] by Chris Maes and Anton Antonov, Wolfram Demonstrations Project.
- Tabulated weights and abscissae with Mathematica source code [9], high precision (16 and 256 decimal places) Legendre-Gaussian quadrature weights and abscissas, for *n*=2 through *n*=64, with Mathematica source code.
- High-precision abscissas and weights for Gaussian Quadrature [10] *n* = 2, ..., 20, 32, 64, 100, 128, 256, 512, 1024 with open source libraries for C/C++ and Matlab.

References

[1] http://apps.nrbook.com/empanel/index.html?pg=179
[2] http://dlmf.nist.gov/3.5.v
[3] http://www.alglib.net/integral/gq/
[4] http://www.gnu.org/software/gsl/
[5] http://ans.hsh.no/home/skk/Publications/Lobatto/PRIMUS_KHATTRI.pdf
[6] http://numericalmethods.eng.usf.edu/topics/gauss_quadrature.html
[7] http://mathworld.wolfram.com/Legendre-GaussQuadrature.html
[8] http://demonstrations.wolfram.com/GaussianQuadrature/
[9] http://processingjs.nihongoresources.com/bezierinfo/legendre-gauss-values.php
[10] http://www.holoborodko.com/pavel/numerical-methods/numerical-integration/

Article Sources and Contributors

Gaussian quadrature *Source*: http://en.wikipedia.org/w/index.php?oldid=493078248 *Contributors*: Advanpix, Araqir, AxelBoldt, Az1568, Berland, Beta16, Brews ohare, Casey Abell, Christos Argyropoulos, Cosmoskramer, Count Iblis, Cygnine, Dneprodzerzhinsk, Dsspiegel, Ejrh, Encyclops, Giftlite, Hankwang, He504, Helder.wiki, Highpriority, Ilmari Karonen, Itub, Jaredwf, Jitse Niesen, Jrvz, Jujutacular, KSmrq, Keenan Pepper, LOL, MSGJ, MarkSweep, Maze-cooperation, Mech Aaron, Mhym, Michael Hardy, Moonliterhythm, Pedrito, Phe, Physics314, Pleasantville, Qiqi.wang, Qwfp, Rafiko77, Salamurai, Scareduck, Segurajj, ServiceAT, Sibi antony, Skarebo, Stevenj, Stevvers, Taxman, Teorth, Tide rolls, TomyDuby, Whitetmac17, Wile E. Heresiarch, 76 anonymous edits

Image Sources, Licenses and Contributors

File:Legendrepolynomials6.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Legendrepolynomials6.svg *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Geek3

License