# Gaussian Quadrature

**Chapter** · January 2020

DOI: 10.1002/9781118445112.stat08242

**3 authors**, including:

Ruohong Li
Indiana University School of Medicine - Lafayette
**6** PUBLICATIONS **141** CITATIONS
SEE PROFILE

Honglang Wang
Indiana University – Purdue University Indianapolis
**23** PUBLICATIONS **640** CITATIONS
SEE PROFILE

# Gaussian Quadrature

Ruohong Li[*], Honglang Wang[†], and Wanzhu Tu[‡]

July 7, 2019

**Abstract**

Statistical analysis often requires evaluation of intractable integrals. Gaussian quadrature is a class of numerical methods for integration. In this article, we review the method of Gaussian quadrature and describe its application in statistics. We also briefly discuss the method's implementation in R and SAS.

**Keywords:** Numerical integration; Gaussian quadrature; Newton-Cotes rules.

## 1 Introduction

Gaussian *quadrature* [stat03875] is a class of numerical methods for approximating the integral $\int_a^b f(x)dx$, by expressing it as a weighted sum of $f(x)$, evaluated at specified points within the range of integration. Gaussian quadrature is one of the most frequently used numerical methods in statistics, with applications ranging from calculating *expectations* [stat05851] and to marginalizing *likelihood* [stat05880] functions [1].

[*] Department of Biostatistics, Indiana University School of Medicine and Fairbanks School of Public Health, Indianapolis, IN 46202, USA; e-mail: rhli@iu.edu.

[†] Department of Mathematical Sciences, Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202, USA; e-mail: hlwang@iupui.edu.

[‡] Department of Biostatistics, Indiana University School of Medicine and Fairbanks School of Public Health, Indianapolis, IN 46202, USA; e-mail: wtu1@iu.edu.

The term *quadrature* stems from the mathematical practice of approximating the area of an irregular shape with small quadrilaterals. In the literature of modern numerical methods, the word quadrature is almost synonymous with *numerical integration*. The method of Gaussian quadrature as we know it today originates from the quadrature rule proposed by *Carl Friedrich Gauss* [stat01301] (1777-1855), who showed that the integral of a polynomial function up to $2n-1$ degrees could be expressed as a sum of $n$ terms exactly [2].

In this article, we review the approximation method of Gaussian quadrature, in contrast to the other frequently used quadrature methods. We start with a general introduction to the Newton-Cotes rules, which preceded the Gaussian rule.

## 2 The Newton-Cotes Rules

A central task of numerical integration is to approximate integrals of the form $\int_a^b f(x)dx$, where $a$ and $b$ can take either finite or infinite values. There are two basic approaches in approximating integrals of this form. One is a method proposed by *Sir Isaac Newton* [stat01326] (1642-1726) and Roger Cotes (1682-1716), commonly referred to as the Newton-Cotes rules or Newton-Cotes formulae [3], which provide a basis for the development of Gaussian quadrature. The other is an approximation proposed by *Pierre Simon Laplace* [stat01315] (1749-1827) in 1774 [4], which we shall not elaborate in the current article.

The basic idea of the Newton-Cotes method is to approximate the integrand with polynomial functions that interpolate $f$ at well-designed nodes within $[a, b]$. The following theorem justifies the approximation [5].

**Theorem 1** (Unisolvence)**.** *Given a set of $n+1$ data points $(x_i, f(x_i))$ where no two $x_i$ are the same, there exists a unique polynomial $p$ of degree at most $n$ with the property*

$$p(x_i) = f(x_i), \qquad i = 0, \cdots, n.$$

Instead of using the usual polynomial basis functions $\{1, x, x^2, \cdots, x^n\}$ to represent the unique $p$, we use the Lagrange basis

$$\phi_j(x) = \prod_{i=0, i\neq j}^{n} \frac{x - x_i}{x_j - x_i}, j = 0, 1, \cdots, n, \tag{1}$$

which satisfies the property $\phi_j(x_i) = \delta_{ij}$; here $\delta_{ij}$ is the Kronecker symbol. The Lagrange polynomials allow for an easy representation of the interpolation polynomial:

$$p(x) = \sum_{i=0}^{n} f(x_i)\phi_i(x). \tag{2}$$

With the interpolation polynomial (2), we have

$$\int_a^b f(x)dx \approx \int_a^b p(x)dx = \sum_{i=0}^{n} f(x_i) \int_a^b \phi_i(x)dx, \tag{3}$$

which is a weighted sum of function values evaluated at $\{x_i\}_{i=0}^n$ with weights

$$w_i := \int_a^b \phi_i(x)dx, \qquad \text{for } i = 0, 1, \cdots, n.$$

The Newton-Cotes rules are special cases of (3), with evenly partitioned $[a, b]$ by nodes $\{x_i = a + ih\}_{i=0}^n$, where $h = (b - a)/n$. These include the well-known *trapezoidal rule* [stat02365] and *Simpson's rule* [stat00751], corresponding respectively to the cases of $n = 1$ and $n = 2$.

**Example 1** (Trapezoidal rule)**.** *With $n = 1$ and evenly distributed nodes $x_0 = a, x_1 = b$, the Lagrange bases are*

$$\phi_0(x) = \frac{x - x_1}{x_0 - x_1} = \frac{x - b}{a - b}, \phi_1(x) = \frac{x - x_0}{x_1 - x_0} = \frac{x - a}{b - a}.$$

*Hence $w_0 = \int_a^b \phi_0(x)dx = (b - a)/2$ and $w_1 = \int_a^b \phi_1(x)dx = (b - a)/2$, which leads to the Trapezoidal rule*

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} f(x_i) \int_a^b \phi_i(x)dx = w_0 f(x_0) + w_1 f(x_1) = \frac{b - a}{2}[f(a) + f(b)].$$

**Example 2** (Simpson's rule)**.** *For $n = 2$ with evenly distributed nodes $x_0 = a$, $x_1 = (b+a)/2$,*

*$x_2 = b$, the Lagrange bases consist of*

$$\phi_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}, \phi_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}, \phi_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}.$$

*Therefore we have $w_0 = \int_a^b \phi_0(x)dx = (b-a)/6$, $w_1 = \int_a^b \phi_1(x)dx = 4(b-a)/6$ and $w_2 = \int_a^b \phi_2(x)dx = (b-a)/6$, and this is the Simpson's rule*

$$\int_a^b f(x)dx \approx \sum_{i=0}^n f(x_i) \int_a^b \phi_i(x)dx = w_0 f(x_0) + w_1 f(x_1) + w_2 f(x_2) = \frac{b-a}{2}[f(a) + 4f(\frac{a+b}{2}) + f(b)].$$

# 3 Gaussian Quadrature and its Variants

Evenly distributed nodes in the Newton-Cotes method do not always lead to accurate and efficient approximations. Finding nodes $\{x_i\}_{i=0}^n$ that produce a more accurate approximation is therefore of great interest. Gaussian quadrature is a method that allows the nodes to be chosen more freely according to the weight function. In comparison with the Newton-Cotes approximation that is exact for polynomial integrands up to $n$ degrees, the Gaussian quadrature rule is exact for polynomial integrands up to $2n+1$ degrees, under $n+1$ suitably chosen nodes. The following examples illustrate the idea.

**Example 3** (Gaussian quadrature with $n = 1$)**.** *Suppose we have a two-point Gaussian quadrature over $[-1, 1]$. The approximation $\int_{-1}^1 f(x)dx \approx \sum_{i=0}^1 f(x_i)w_i dx$ has 4 unknown quantities: Two nodes $\{x_0, x_1\}$,and two weights $\{w_0, w_1\}$. The Gaussian quadrature approximation is exact for any polynomials $f$ up to $2n + 1 = 3$ degrees: $\int_{-1}^1 1dx = w_0 + w_1 = 2, \int_{-1}^1 xdx = w_0x_0 + w_1x_1 = 0, \int_{-1}^1 x^2dx = w_0x_0^2 + w_1x_1^2 = 2/3, \int_{-1}^1 x^3dx = w_0x_0^3 + w_1x_1^3 = 0$. This leads to solutions $w_0 = 1, w_1 = 1, x_0 = -1/\sqrt{3}, x_1 = 1/\sqrt{3}$. Hence, a Gaussian Quadrature approximation with $n = 1$ is*

$$\int_{-1}^1 f(x)dx \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right).$$

For a Gaussian quadrature over the interval $[-1, 1]$ with a given $n$, the solutions to the $2n+2$ equations $\sum_{i=0}^{n} w_i x_i^k = \int_{-1}^{1} x^k dx, k = 0, 1, \cdots, 2n+1$ are given as follows. See Chapter 5 of Givens and Hoeting (2005) for further details [6].

**Theorem 2.** *Let $p_n$ be the Legendre polynomial of degree $n$ defined on $[-1, 1]$ such that $\int_{-1}^{1} p_n(x)p_m(x)dx = \delta_{mn}$ for all $m \neq n$. Denote the $n+1$ roots of Legendre polynomial $p_{n+1}$ as $\{x_i : i = 0, \cdots, n\}$. Then for any polynomial $p$ up to $2n+1$ degrees, we have*

$$\int_{-1}^{1} p(x)dx = \sum_{i=0}^{n} p(x_i) \int_{-1}^{1} \phi_i(x)dx. \tag{4}$$

Hence a Gaussian quadrature rule over the interval $[-1, 1]$ with a given $n$ is

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=0}^{n} f(x_i) \int_{-1}^{1} \phi_i(x)dx, \tag{5}$$

where $\{x_i : i = 0, \cdots, n\}$ are the roots of a Legendre polynomial of degree $n$, and $\{\phi_i(x)\}_{i=0}^{n}$ are the Lagrange basis functions given in (1).

**Example 4** (Gaussian quadrature with $n = 1$)**.** *The Legendre polynomials of degree $\leq 3$ are $p_0(x) = 1, p_1(x) = x, p_2(x) = \frac{3}{2}(x^2 - \frac{1}{3}), p_3(x) = \frac{5}{2}(x^3 - \frac{3}{5})$. With $n = 1$, the roots of $p_2$ are $x_0 = -1/\sqrt{3}, x_1 = 1/\sqrt{3}$ and $w_0 = \int_{-1}^{1} \phi_0(x)dx = \int_{-1}^{1} \frac{x-x_1}{x_0-x_1}dx = 1, w_1 = \int_{-1}^{1} \phi_1(x)dx = \int_{-1}^{1} \frac{x-x_0}{x_1-x_0}dx = 1$. This is consistent with the result obtained in Example 3.*

More generally, Gaussian quadrature treats the integral of $f(x)$ as the *expectation* of $f^*(x) = \frac{f(x)}{w(x)}$ with $w(x)$ as a *probability density function* [stat03900] with $\int_a^b x^k w(x)dx[\infty$ for all $k \geq 0$. The integral $\int_a^b f(x)dx$ therefore can be approximated by

$$\int_a^b f(x)dx = \int_a^b \frac{f(x)}{w(x)} w(x)dx = \int_a^b f^*(x)w(x)dx \approx \sum_{i=1}^{n} f^*(x_i)w_i,$$

where $x_i$ are the roots of some orthogonal polynomials determined by $w(x)$ and the weights $w_i$ are calculated based on these $x_i$. Davis and Robinowitz (2007) presented the following result for the general Gaussian quadrature [7].

**Theorem 3.** *Let $p_{n+1}$ be a nontrivial polynomial of degree $n+1$ such that it is orthogonal* *to* $\{1, x, \cdots, x^n\}$ *with respect to* $w(x)$*, i.e.,*

$$\int_a^b w(x)x^k p_{n+1}(x) = 0, \forall k = 0, 1, \cdots, n.$$

*If we pick the* $n+1$ *nodes* $\{x_i : i = 0, \cdots, n\}$ *to be the roots of* $p_{n+1}$*, then there exist* $n+1$ *weights* $\{w_i : i = 0, \cdots, n\}$ *which make the Gaussian quadrature exact, i.e.,*

$$\int_{-1}^{1} p(x)w(x)dx = \sum_{i=0}^{n} p(x_i)w_i \tag{6}$$

*for any polynomial* $p$ *of degree less than or equal to* $2n+1$*.*

Computational details of the quadrature nodes and weights can be found in Abramowitz and Stegun's *Handbook of Mathematical Functions* [3]. In Table 1 below, we summarize some of the most commonly used Gaussian quadrature rules for different integral regions and weight functions $w(x)$.

- Insert Table 1 here.

# 4 Extensions of Gaussian Quadrature Rules

## 4.1 The Composite Quadrature Rule

The error of the Gaussian quadrature approximation of $\int_a^b f dx$ on $[a, b]$ depends on the length of $[a, b]$. Sometimes it is preferable to divide $[a, b]$ into smaller subintervals, and use quadrature rules to compute the integral $\int f dx$ on each subinterval. The final approximation of $\int_a^b f(x)dx$ is obtained by adding results from all subintervals. Such a scheme is referred to as the composite quadrature method.

## 4.2 Multivariate Integration

In statistics, integration over multi-dimensional domains is quite common. In many situations, a multivariate integration can be expressed as a sequence of nested univariate integrals, on which univariate quadrature rules can be applied, and the final approximation assembled through a *tensor* [stat02360] product. Cools (2002) provided more computational details [8].

To illustrate the idea, we consider a two-dimensional integral. We first note that

$$\int_a^b \int_c^d f(x,y)dydx = \int_a^b g(x)dx,$$

where $g(x) = \int_c^d f(x,y)dy$. With univariate quadrature rules, we write

$$\int_a^b \int_c^d f(x,y)dydx = \int_a^b g(x)dx \approx \sum_{i=0}^n g(x_i)w_i.$$

For each $x_i$, we have $g(x_i) = \int_c^d f(x_i,y)dy \approx \sum_{j=0}^m f(x_i,y_j)\widetilde{w}_j$.

The multivariate integral, therefore, can be approximated by

$$\int_a^b \int_c^d f(x,y)dydx = \int_a^b g(x)dx \approx \sum_{i=0}^n g(x_i)w_i \approx \sum_{i=0}^n \sum_{j=0}^m f(x_i,y_j)\widetilde{w}_j w_i.$$

For a general multivariate integration, suppose we have an integrand function $f : \Omega \to \mathbb{R}$ and a weight function $w : \Omega \to \mathbb{R}$ with $q$ integration variables $x_1, \cdots, x_q$ varying on the domain $\Omega \subset \mathbb{R}^q$. We express the multivariate integral $\int_\Omega f(\mathbf{x})w(\mathbf{x})d\mathbf{x}$ as a sequence of univariate integrals nested through a tensor product,

$$Q[f] := (Q_1^{(1)} \otimes ... \otimes Q_1^{(q)})[f] := \sum_{l_1=0}^{L_1} ... \sum_{l_q=0}^{L_q} [f(x_{l_1}^{(1)}, ..., x_{l_q}^{(q)}) \prod_{k=1}^q w_{l_k}^{(k)}],$$

under the assumptions:

1. $w(\boldsymbol{x}) = w(x_1, ..., x_q)$ can be written as $w(\boldsymbol{x}) = \prod_{k=1}^q w_k(x_k)$.

2. The domain of integration $\Omega = \Omega_1 \times ... \times \Omega_q$, where $\Omega_k \subset R, k = 1, ..., q$.

Note that for each $k \in \{1, 2, \cdots, q\}$, the nodes and weights of the corresponding univariate quadrature rules are $\{x_{l_k}^{(k)} : l_k = 0, 1, \cdots, L_k\}$ and $\{w_{l_k}^{(k)} : l_k = 0, 1, \cdots, L_k\}$, respectively.

Multivariate integration, however, is a computationally intensive process as its structural complexity and computational intensity increase exponentially with the dimension. Analysts often face the "*curse of dimensionality*" [stat00408] in higher dimensional situations. For example, with $q = 7$ variables in an integral and $L = 20$ nodes for each dimension, the total number of nodes will be $20^7 = 1,280,000,000$. As a result, the tensor product of univariate Newton-Cotes rules or Gaussian quadrature rules is rarely applicable in high-dimensional integration. Instead, lattice rules [9] and sparse-grid integration techniques [10; 11] are often used. Another obvious limitation of the full-grid approximation is that when the variables are correlated, nested, or crossed, it will be impossible to express the multivariate integral as a sequence of one-dimensional integrals. In such situations, Monte-Carlo and quasi-Monte-Carlo techniques [12; 13] and *Laplace's method* [stat02222] [14] are more frequently employed.

## 4.3 Adaptive Quadrature

The number of nodes often affects the accuracy of a quadrature approximation, so determining the number of nodes to achieve a prespecified error tolerance is of great interest. Adaptive quadrature is a sequential technique that is designed to address the issue. An adaptive quadrature approximation has the following steps:

S1 Use a quadrature rule: $Q =\mathsf{Q}(f, a, b, \tau)$ to approximate $I(f) = \int_a^b f(x)dx$, where $\tau$ is a required tolerance.

S2 Evaluate the error $\varepsilon \approx |Q - I(f)|$

S3 If $\varepsilon[\tau$, then stop; Otherwise let $m = (a + b)/2$ and approximate $I(f)$ by

$$Q = \mathsf{Q}(f, a, m, \tau/2) + \mathsf{Q}(f, m, b, \tau/2).$$

S4 Repeat S2 and S3 for each of the $\mathsf{Q}(f, a, m, \tau/2)$ and $\mathsf{Q}(f, m, b, \tau/2)$ until the stopping criterion is met.

# 5 A Numerical Study

We present a small study to highlight the performance differences among the most frequently used quadrature methods. All analyses were performed using the $R$ [stat05033t] software[15].

Using $\int_0^\pi x\sin(x)dx = \pi$ as an example, we compare the accuracy of the previously discussed methods for a fixed number of nodes and evaluate the corresponding computing time with a given error tolerance.

Figure 1 shows the results from the (A) composite trapezoidal rule, (B) composite Simpson's, rule, and (C) Gauss-Legendre rule with $n = 2$. Table 2 shows the accuracy of different approximation methods under the same number of nodes. Table 3 summarizes the computing time of adaptive Trapezoidal rule, adaptive Simpson's rule, and adaptive Gauss-Legendre rule to achieve different error tolerance levels on a machine with Windows 10 Home 64, two 16GB DDR4 2400 SODIMM, Core i7-7700 Processor (8MB Cache, up to 4.20GHz), and four cores.

- Insert Figure 1 here.

- Insert Tables 2-3 here.

While error decreases with an increased number of nodes in all methods, Gauss-Legendre quadrature and Newton-Cotes rules appear to outperform the other methods, including the composite trapezoidal rule, composite Simpson's rule, and Gauss-Chebyshev quadrature, especially with a larger number of nodes. For a given level of error tolerance, the computing time (in seconds) varies among the methods. The adaptive Simpson's rule is the fastest to converge, while the adaptive Gauss-Legendre quadrature converges slightly slower than the adaptive Simpson's rule, but is still much faster than the adaptive trapezoidal rule.

# 6 Statistical Applications

Gaussian quadrature's relevance and importance to statistics have been recognized, mainly in areas where numerical integration is required, to calculate expectation and performing marginalization (i.e., to integrate out *random effects* [stat07558]). In this section, we briefly describe the statistical modeling contexts in which Gaussian quadrature methods are used, all through a simple example of mixed-effect logistic regression.

Suppose we have repeatedly measured data from $n$ independent subjects $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1}^n$, where $\boldsymbol{y}_i = (y_{i1}, ..., y_{it_i})^T$ is the outcomes from the $i$th subject, and $\boldsymbol{x}_i = (x_{i1}, ..., x_{it_i})^T$ is the vector of independent variables from the same subject. Here we assume that within each subject, the responses are correlated, and the within-subject correlation depends solely on the shared random effect $\gamma_i$. To model, we write $y_{ij}|(x_{ij}, \gamma_i) \sim \text{Bernoulli}(p_{ij})$ with

$$\log\left(\frac{p_{ij}}{1-p_{ij}}\right) = x_{ij}\beta + \gamma_i,$$

where the random effect $\gamma_i \overset{i.i.d.}{\sim} N(0, \sigma^2)$.

**Marginalization** Estimates of the fixed effects $\beta$ and $\sigma^2$ of the above model can be obtained either through the *maximum likelihood estimator* (MLE) [stat01663] or the *restricted maximum likelihood* (REML) [stat05953] estimator, based on the marginalized likelihood:

$$L(\beta, \sigma^2) = \prod_{i=1}^n \int f(\gamma_i|\sigma^2) f(\boldsymbol{y}_i|\beta, \gamma_i) d\gamma_i = \prod_{i=1}^n \int f(\gamma_i|\sigma^2) \prod_{j=1}^{t_i} f(y_{ij}|\gamma_i, \beta) d\gamma_i.$$

where $f(y_{ij}|\gamma_i, \beta) = p_{ij}^{y_{ij}}(1-p_{ij})^{1-y_{ij}}$ and $f(\gamma_i|\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{\gamma_i^2}{2\sigma^2})$, and $p_{ij} = \frac{\exp(x_{ij}\beta+\gamma_i)}{1+\exp(x_{ij}\beta+\gamma_i)}$.

Maximization of the likelihood requires evaluation of the function for given $\beta$ and $\sigma^2$. All terms involved in the above product are integrals, to which quadrature rules can be applied for evaluation.

**Expectation** An iterative expectation-maximization (*EM*) *algorithm* [stat00410] [16] can also be used for estimating $\beta$ and $\sigma^2$. In the previously described model, the conditional log-likelihood is

$$\ell(\beta, \sigma|\boldsymbol{y}, \gamma) = \sum_{i=1}^{n} \log\left\{f(\boldsymbol{y}_i|\gamma_i, \beta) f(\gamma_i|\sigma)\right\}.$$

With $\theta^{(r)}$ as the estimate from the $r$-th iteration for $\theta := (\beta, \sigma^2)$, at the $(r+1)$th iteration, one calculates the expectation of the log-likelihood with respect to the current conditional distribution $f(\gamma|\boldsymbol{y}, \theta^{(r)})$. This step is often called the E-step:

$$Q(\theta|\theta^{(r)}) = E_{\gamma|\boldsymbol{y},\theta^{(r)}}\{\ell(\beta, \sigma|\boldsymbol{y}, \gamma)\} = \sum_{i=1}^{n} E_{\gamma_i|\boldsymbol{y}_i,\theta^{(r)}} \log\left\{f(\boldsymbol{y}_i|\gamma_i, \beta) f(\gamma_i|\sigma)\right\},$$

where the expectations $E_{\gamma_i|\boldsymbol{y}_i,\theta^{(r)}} \log\left\{f(\boldsymbol{y}_i|\gamma_i, \beta) f(\gamma_i|\sigma)\right\}$ are with respect to the conditional distribution,

$$f(\gamma_i|\boldsymbol{y}_i, \theta^{(r)}) \propto f(\boldsymbol{y}_i, \gamma_i|\theta^{(r)}) = f(\boldsymbol{y}_i|\gamma_i, \theta^{(r)}) f(\gamma_i|\theta^{(r)}).$$

In the maximization, or M-step, we find the value $\theta^{(r+1)}$ that maximizes the $Q(\theta|\theta^{(r)})$,

$$\theta^{(r+1)} = \underset{\theta}{\operatorname{argmax}}\, Q(\theta|\theta^{(r)}).$$

The EM estimates are achieved by repeating the E and M steps until convergence. Gaussian quadrature rules can be used to evaluate the expectations in the E-step:

$$E_{\gamma_i|\boldsymbol{y}_i,\theta^{(r)}} \log\left\{f(\boldsymbol{y}_i|\gamma_i, \beta) f(\gamma_i|\sigma)\right\} = \int log\, f(y, \gamma|\theta) f(\gamma|y, \theta^{(r)}) d\gamma = \frac{\int log(f(y,\gamma|\theta)) f(y, \gamma|\theta^{(r)}) d\gamma}{\int f(y, \gamma|\theta^{(r)}) d\gamma}$$

.

# 7 Software Implementation

There are multiple R packages that can be used for approximating definite integrals. For example, the function area from the package MASS calculates an integral in a finite range by using an adaptive Simpson's rule. Function integrate from the package stats evaluates integrals on infinite ranges by using an adaptive Gauss-Kronrod quadrature method. Function ghQuad from the package fastGHQuad implements Gauss-Hermite quadrature rules with high degree polynomials. And function aghQuad from the package fastGHQuad performs approximation by using an adaptive Gauss-Hermite quadrature. For multi-dimensional integration, eval.quad from the package MultiGHQuad offers an approximation with the Gauss-Hermite quadrature rule.

Gaussian quadrature methods are also used in statistical modeling settings, especially in *mixed effect models* [stat07549]. For implementation of generalized linear mixed effect models, SAS PROC GLIMMIX and R function glmer (package lme4) are popular tools. The R function glmer can fit a single-level *generalized linear mixed model* [stat07540] without random slopes by using an adaptive Gaussian quadrature rule [17]. For more sophisticated models involving multiple levels of random effects, the function resorts to the Laplace approximation. Option nAGQ= in glmer allows users to specify the number of quadrature points. When nAGQ=1, Laplace method is used. Similarly, SAS PROC GLIMMIX allows users to choose adaptive Gaussian-Hermite quadrature for integral approximation through option METHOD = QUAD, and to specify the number of quadrature points with the qpoints option. One should refer to their respective user manuals for technical details.

## 8 Discussion

Gaussian quadrature is a powerful tool for numerical integration. A fundamental property of this method is that it yields exact values of integrals for polynomials up to $2n-1$ degrees, in contrast to the Newton-Cotes' optimality for polynomial functions up to $n$ degrees. In statistics, it is one of the most frequently used numerical methods. Its main use includes calculation of expectations and marginalization of likelihood functions. Numerical performance of Gaussian quadrature is generally efficient and stable. This said, the appropriateness of a numerical integration method in a specific application almost always depends on the nature of the integrand function and the range of the integral. For smooth and well-behaved functions, the choice of different numerical methods may not be as consequential. But for less well-behaved functions, attention and care are always needed to assure good accuracy and fast convergence.

# References

[1] Brass, H., Fischer, J.-W. & Petras, K. *The Gaussian quadrature method* (Institute für Mathematik, Techn. Univ., 1997).

[2] Stoer, J. & Bulirsch, R. *Introduction to numerical analysis*, vol. 12 (Springer Science & Business Media, 2013).

[3] Abramowitz, M., Stegun, I. A. *et al. Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, vol. 55 (Dover publications New York, 1972).

[4] Laplace, P. S. Memoir on the probability of the causes of events. tome sixième. *Mémoires de Mathématique et de Physique (English translation by SM Stigler 1986. Statist. Sci., 1 (19): 364-378)* (1774).

[5] Waring, E. Vii. problems concerning interpolations. *Philosophical Transactions of the Royal Society of London* 59–67 (1779).

[6] Givens, G. H. & Hoeting, J. A. *Computational Statistics* (2005).

[7] Davis, P. J. & Rabinowitz, P. *Methods of numerical integration* (Courier Corporation, 2007).

[8] Cools, R. Advances in multidimensional integration. *Journal of Computational and Applied Mathematics* **149**, 1–12 (2002).

[9] Sloan, I. H. & Joe, S. *Lattice methods for multiple integration* (Oxford University Press, 1994).

[10] Zenger, C. *Sparse grids* (Technische Universität, 1990).

[11] Garcke, J. & Griebel, M. *Sparse grids and applications*, vol. 88 (Springer Science & Business Media, 2012).

[12] Caflisch, R. E. Monte carlo and quasi-monte carlo methods. *Acta numerica* **7**, 1–49 (1998).

[13] Niederreiter, H. *Random number generation and quasi-Monte Carlo methods*, vol. 63 (Siam, 1992).

[14] Wong, R. *Asymptotic approximations of integrals*, vol. 34 (SIAM, 2001).

[15] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria (2013). URL http://www.R-project.org/.

[16] Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)* 1–38 (1977).

[17] Pinheiro, J. C. & Chao, E. C. Efficient laplacian and adaptive gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics* **15**, 58–81 (2006).

Table 1: Variants of Gaussian quadrature rules with different integration regions and weight functions.

| Name (Density) | Interval | w(x) | $x_i$ are roots of |
|---|---|---|---|
| Legendre (Uniform) | [-1,1] | 1 | Legendre polynomials |
| Jacobi (Beta) | (-1,1) | $(1-x)^\alpha x^\beta$ | Jacobi polynomials |
| Chebyshev (First kind) | (-1,1) | $1/\sqrt{1-x^2}$ | Chebyshev polynomials, first kind |
| Chebyshev (Second kind) | [-1,1] | $\sqrt{1-x^2}$ | Chebyshev polynomials, second kind |
| Laguerre (Exponential) | $[0,\infty)$ | $\exp(-x)$ | Laguerre polynomials |
| Generalized Laguerre (Gamma) | $[0,\infty)$ | $x^\alpha \exp(-x)$ | Generalized Laguerre polynomials |
| Hermite (Normal) | $(-\infty,\infty)$ | $\exp(-x^2)$ | Hermite polynomials |

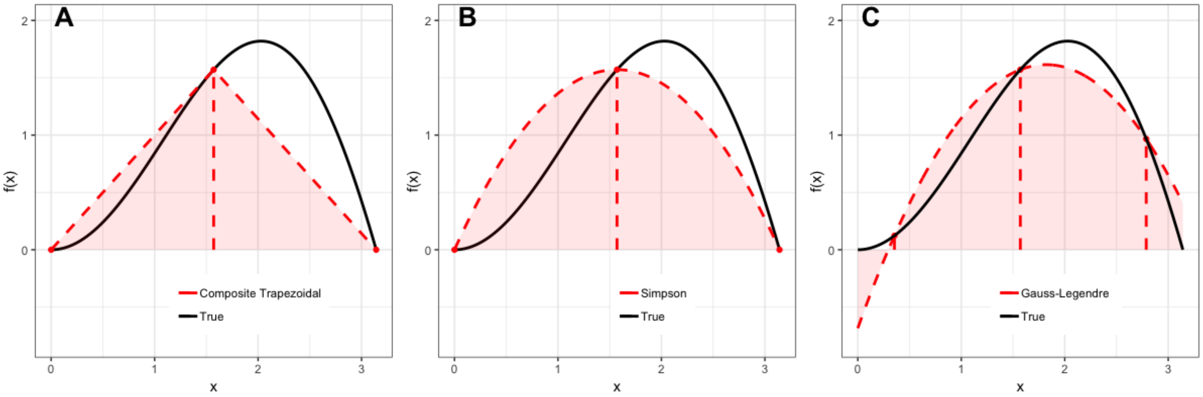
Table 2: Approximation errors of different numerical quadrature methods in Section 5.

| | Number of Nodes | | |
|---|---|---|---|
| Method | 3 | 7 | 11 |
| Newton-Cotes | 0.1483 | $2.7891 \times 10^{-5}$ | $1.8013 \times 10^{-10}$ |
| Composite Trapezoidal | 0.6742 | 0.0721 | 0.0259 |
| Composite Simpson's | 0.1483 | 0.0014 | 0.0002 |
| Gauss-Chebyshev | 0.7079 | 0.0022 | 0.0003 |
| Gauss-Legendre | 0.1008 | $8.2110 \times 10^{-10}$ | $4.4409 \times 10^{-15}$ |

Table 3: Computing time in seconds of different numerical quadrature methods in Section 5 to achieve the pre-specified approximation error tolerance.

| | Error Tolerance | | |
|---|---|---|---|
| Method | $10^{-4}$ | $10^{-6}$ | $10^{-10}$ |
| Adaptive Trapezoidal | 0.02 | 1.00 | 94.64 |
| Adaptive Simpson's | 0.01 | 0.02 | 0.06 |
| Adaptive Gauss-Legendre | 0.02 | 0.05 | 0.33 |

Figure 1: The interpolation polynomials for various quadrature rules in Section 5.


A
B
C
f(x)
x
Composite Trapezoidal
Simpson
Gauss-Legendre
True