The Discrete Moment Method for the numerical integration of piecewise higher order convex functions

CrossMark

András Prékopa, Mariya Naumova *

Piscataway, NJ, United States

ABSTRACT

A new numerical integration method, termed Discrete Moment Method, is proposed for univariate functions that are piecewise higher order convex. This means that the interval where the function is defined can be subdivided into non-overlapping subintervals such that in each interval all divided differences of given orders, do not change the sign. The new method uses piecewise polynomial lower and upper bounds on the function, created in connection with suitable dual feasible bases in the univariate discrete moment problem and the integral of the function is approximated by tight lower and upper bounds on them. Numerical illustrations are presented for the cases of the normal, exponential, gamma and Weibull probability density functions.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Numerical integration methods generally work in such a way that the integrand is evaluated at a finite number of points, called integration points or base points, and a weighted sum of these values approximates the integral. The base points and weights depend on the specific method used and the required accuracy.

An important part of the analysis of any numerical integration method is the study of the approximation error as a function of the number of integrand evaluations. A method which yields a small error for a small number of evaluations is usually considered efficient.

The scientific literature is replete with the techniques of numerical integration. For example, Davis, Rabinowitz and Engels in [2,3] list about a thousand papers on the topic. Many integration rules (see, e.g., [1,6,13]) use interpolation functions, typically polynomials, which are easy to integrate. The simplest rules of this type are the midpoint (or rectangle), the trapezoidal and the Simpson's rules, where for a small interval $[a, b]$ the approximations

$$\int_a^b f(x)dx \approx (b-a)f\left(\frac{a+b}{2}\right), \tag{1.1}$$

$$\int_a^b f(x)dx \approx (b-a)\frac{f(a)+f(b)}{2}, \tag{1.2}$$

$$\int_a^b f(x)dx \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right), \tag{1.3}$$

respectively, are used. These methods rely on a "divide and conquer" strategy, whereby an integral on a relatively large set is broken down into integrals on smaller sets.

---

* Corresponding author.
   E-mail address: mariya.v.naumova@gmail.com (M. Naumova).

Interpolation with polynomials evaluated at equally-spaced points in $[a, b]$ yields the Newton–Cotes formulas, of which the rectangle and the trapezoidal rules are examples. Simpson's rule, which is based on a polynomial of order 2, is also a Newton–Cotes formula. If we allow the intervals between interpolation points to vary in length, we find other integration formulas, such as the Gaussian quadrature formulas. A Gaussian quadrature rule is typically more accurate than a Newton–Cotes rule which requires the same number of function evaluations, if the integrand is smooth. For a large number of variants of Gaussian quadrature the reader is referred to [4].

Romberg's method is based upon the approximation of the integral by the trapezoidal rule. Quadrature formulas of higher error order are produced by successive division of the step size by 2 and by an appropriate linear combination of the resulting approximations for the integral.

First, one partitions $[a, b]$ into N_0 subintervals of length $h_0 = (b - a)/N_0$ and sets

$$N_i = 2^i N_0, \qquad h_i = h_0/2^i, \quad i = 0, 1, \ldots,$$

then the integral is expressed as

$$\int_a^b f(x)dx = L_i^{(k)}(f(x)) + O(h_i^{2(k+1)}),$$

where $L_i^{(k)}(f(x))$ is a quadrature formula with error order $O(h_i^{2(k+1)})$.

Romberg's method provides us with accurate results if the integrand has multiple continuous derivatives, though fairly good results may be obtained if only a few derivatives exist. We also mention numerical methods by [14] that are useful when it is impossible or undesirable to use derivatives of the integrand.

In this paper we propose a new univariate numerical integration method. We create lower and upper bounding polynomials for the function on a finite grid but ensure that the integrals of the bounding polynomials provide us with tight lower and upper bounds for the integral of our function in an entire interval. We use Lagrange polynomials for bounding that are natural outcomes of the use of the discrete power moment problem. We illustrate our new method for the functions: $e^{-x^2/2}, x^m e^{-x^2/2}, \left(\frac{x}{\lambda}\right)^{m-1} e^{-\left(\frac{x}{\lambda}\right)^m}$, and $\lambda e^{-\lambda x}$.

2. Bounding by Lagrange polynomials

2.1. Summary of the discrete moment problem

In what follows we assume the knowledge of the elements of linear programming. A brief summary of it can be found in [9,10]. The material in this section is based on [7].

Consider the following linear programming problems:

$$
\begin{aligned}
\min (\max) \quad & \sum_{i=0}^{n} f(z_i)p_i \\
\text{subject to} \quad & \sum_{i=0}^{n} z_i^k p_i = \mu_k, \quad k = 0, \ldots, m, \\
& p_i \geq 0, \quad i = 0, \ldots, n.
\end{aligned}
\tag{2.1}
$$

Problem (2.1) is called the discrete power moment problem. Here $f(z)$ is a discrete function, $m < n$, and $\{p_i\}$ are the decision variables that form a probability distribution for the random variable X that has support set $\{z_0, \ldots, z_n\}$.

Let the matrix of the equality constraints, its columns, and the right hand side vector be designated by $\mathbf{A}, \mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_n$ and $\mathbf{b}$, respectively. Thus, we can rewrite the power moment problem in the following form:

$$
\begin{aligned}
\min (\max) \quad & \sum_{i=0}^{n} f_i x_i \\
\text{subject to} \quad & \sum_{i=0}^{n} \mathbf{a}_i x_i = \mathbf{b}, \\
& x_i \geq 0, \quad i = 0, \ldots, n,
\end{aligned}
\tag{2.2}
$$

where

$$
\mathbf{a}_i = \begin{pmatrix} 1 \\ z_i \\ z_i^2 \\ \vdots \\ z_i^m \end{pmatrix}, \quad i = 0, 1, \ldots, n; \qquad \mathbf{b} = \begin{pmatrix} 1 \\ \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{pmatrix}.
\tag{2.3}
$$