Stable high-order quadrature rules with equidistant points

Daan Huybrechs

Report TW 528, September 2008

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Stable high-order quadrature rules with equidistant points

Daan Huybrechs

Report TW 528, September 2008

Department of Computer Science, K.U.Leuven

Abstract

Newton-Cotes quadrature rules are based on polynomial interpolation in a set of equidistant points. They are very useful in applications where sampled function values are only available on a regular grid. Yet, these rules rapidly become unstable for high orders. In this paper we review two techniques to construct stable high-order quadrature rules using equidistant quadrature points. The stability follows from the fact that all coefficients are positive. This result can be achieved by allowing the number of quadrature points to be larger than the polynomial order of accuracy. The computed approximations then implicitly correspond to the integral of a least squares approximation of the integrand. We show how the underlying discrete least squares approximation can be optimised for the purpose of numerical integration.

Keywords : numerical integration, least squares approximation, discrete orthogonal polynomials
AMS(MOS) Classification : Primary : 41A55, Secondary : 65F20, 42C05

Stable high-order quadrature rules with equidistant points

Daan Huybrechs*†

Abstract

Newton-Cotes quadrature rules are based on polynomial interpolation in a set of equidistant points. They are very useful in applications where sampled function values are only available on a regular grid. Yet, these rules rapidly become unstable for high orders. In this paper we review two techniques to construct stable high-order quadrature rules using equidistant quadrature points. The stability follows from the fact that all coefficients are positive. This result can be achieved by allowing the number of quadrature points to be larger than the polynomial order of accuracy. The computed approximations then implicitly correspond to the integral of a least squares approximation of the integrand. We show how the underlying discrete least squares approximation can be optimised for the purpose of numerical integration.

1 Introduction

A recurring problem in computational science is the approximate evaluation of the integral

$$I[f] := \int_a^b w(x) f(x)\,\mathrm{d}x \approx Q[f] := \sum_{j=1}^N w_j f(x_j) \tag{1}$$

by a quadrature $Q[f]$ rule with N points x_i and weights w_i. There is a rich body of literature on numerical integration, we refer the reader to the volumes [14, 6, 7] and the references therein for a broad overview.

A case of great practical importance is the case where f is known only in equidistant points, i.e., in the points

$$x_j = a + (j-1)\frac{b-a}{N-1}, \qquad j = 1, \ldots, N. \tag{2}$$

Popular quadrature rules in this setting are the (composite) trapezoidal rule and Simpson rules. These are low-order variants of the family of Newton-Cotes rules, which for the set of N points (2) are exact for all polynomials up to degree $N-1$. We say that the rule has order N. Newton-Cotes rules are easy to apply and easy to implement for a variety of weight functions $w(x)$, but the low-order

*Department of Computer Science, Katholieke Universiteit Leuven, Belgium (daan.huybrechs@cs.kuleuven.be)

†The author is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO).

rules converge slowly and the high-order rules are numerically unstable. We will discuss these properties in more detail in §2.

Stability in numerical integration follows from having positive weights. It is well-known that quadrature rules with positive weights converge for all continuous functions f on $[a, b]$. A fundamental theorem in numerical integration, due originally to Tchakaloff [15] (see also [14, 4]), states that for any functional of the form (1) with a positive weight function $w(x) > 0$, a quadrature rule of order d exists with d points and with positive weights. When searching for a rule with positive weights, Tchakaloff's result essentially supplies an upper bound on the required number of points. The theorem is more general than stated here and holds for a variety of basis functions and for higher dimensions. For the case of polynomials and univariate integrals, the problem of determining such rules is completely resolved by the existence of Guassian quadrature or Clenshaw-Curtis quadrature. These rules however do not have equidistant points. The problem in higher dimensions is much more challenging [2].

It turns out that rules with positive weights can be constructed on equidistant grids by letting the number of quadrature points N be larger than the order d of the rule. It was shown by Davis that if a rule of order d with positive weights exists on N equidistant points, then another rule of order d exists with only d nonzero weights on the same grid [4]. This rule thus achieves Tchakaloff's upper bound. It was shown later by Wilson that the minimal number of equidistant points N scales as $N \sim Cd^2$ as $d \to \infty$ [18]. The constant C is fairly modest, we illustrate numerically in this paper that $C \approx 0.07$ for the case $w(x) = 1$. For example, a quadrature rule of order 20 requires only $N = 33$. Conversely, when samples are given on an equidistant grid with N points, a quadrature rule of order $d \approx \frac{1}{0.07}\sqrt{N}$ with positive weights may be used to integrate the function.

A first technique to construct quadrature rules with positive coefficients is by solving a least squares problem. The connection between quadrature rules and discrete least squares problems was also examined by Wilson [17]. It appears that this connection has not been further explored in a systematic manner since the publication in 1970 of [17, 18]. On the other hand it has long been, and still is, quite common to construct a (discrete) least squares approximation of a (sampled) function. This approximation can then be integrated exactly. Both approaches are obviously related: in the polynomial case, the order of the quadrature rule corresponds to the polynomial degree of the least squares approximation. We intend to show in this paper that the interpretation as a quadrature rule does have some advantages. First, the requirement that the weights should be positive yields a natural and generally applicable stopping criterion for discrete least squares approximations. Increasing the degree beyond a certain value may yield numerical instability for certain functions. Second, the connection with least squares problems supplies a numerically stable and efficient way to construct Newton-Cotes quadrature rules and general interpolatory quadrature rules on arbitrarily spaced points. We note that such rules are most often constructed by solving an ill-conditioned system with a Vandermonde-like matrix. However, explicit expressions for the weights can easily be derived in terms of discrete orthogonal polynomials. Third, discrete least squares approximations for the sole purpose of numerical integration benefit from optimized choices of a weighted discrete inner product. The weight factors themselves are related to numerical integration.

The described algorithm to construct least squares quadrature rules is fast and stable and rules of very high order can in applications be computed on the fly. The algorithm as presented in §4 was suggested in the context of quadrature rules in [17], but the method for discrete least squares problems using recurrences for orthogonal polynomials dates back to Forsythe [8].

A second technique to construct quadrature rules with positive coefficients is by careful selection of the points in interpolatory quadrature. The construction of a quadrature rule with only d positive weights, corresponding to d out of N possible equidistant points, can be achieved by solving a least squares problem subject to linear inequality constraints. A convergent algorithm for this particular type of problem was proposed in [12], called the NNLS algorithm (nonnegative least squares). The result is a class of interpolatory quadrature rules with guaranteed numerical stability and convergence for increasing order.

We continue the paper in §2 with an illustration of the difficulties of using Newton-Cotes quadrature. We describe least squares quadrature rules in §3, a stable implementation based on the method of Forsythe in §4 and nonnegative least squares methods in §5. We end the paper with numerical results in §6.

We would like to stress the fact that most of the theory in §2 is present already in the papers [17, 18]. In this paper, we supplement a self-contained description of this theory with pointers to and descriptions of applicable existing algorithms and with extensive numerical examples in the later sections.

2 Newton-Cotes quadrature

2.1 Preliminaries

In the following, we will characterize the quadrature rule

$$Q[f] := \sum_{j=1}^{N} w_j\, f(x_j), \tag{3}$$

with the vector of quadrature points $\mathbf{x} = [x_j]^T \in \mathbb{R}^N$ and the weight vector $\mathbf{w} = [w_j]^T \in \mathbb{R}^N$ by $[\mathbf{x}, \mathbf{w}]$. We are interested in a rule that is exact on a finite-dimensional function space

$$V := \operatorname{span}\{\phi_i\}_{i=0}^{d-1}. \tag{4}$$

For a given vector $\mathbf{x}$, interpolatory quadrature rules in general require $N = d$ quadrature points to satisfy the d exactness conditions

$$Q[\phi_i] = I[\phi_i], \qquad i = 0, \ldots, d-1. \tag{5}$$

The corresponding weights can be found by solving the linear system

$$S\mathbf{w} = B, \tag{6}$$

where the matrix $S \in \mathbb{R}^{d\times d}$ consists of function evaluations of ϕ_i,

$$S_{i,j} = \phi_i(x_j), \tag{7}$$

and the right hand side vector $B \in \mathbb{R}^d$ contains the *moments*

$$B_i = I[\phi_i]. \tag{8}$$

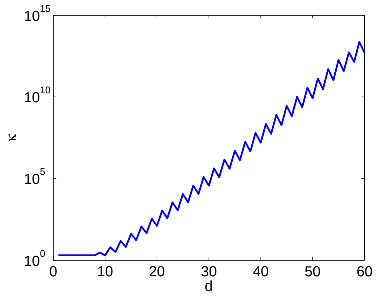


Figure 1: The quantity $\kappa(\mathbf{w})$ for Newton-Cotes quadrature as a function of the order d. High-order rules rapidly become very unstable.

The quadrature rule $[\mathbf{x}, \mathbf{w}]$ constructed in this way interpolates the function f in the points $\mathbf{x}$ by a linear combination of the basis functions and integrates the result exactly, hence the name interpolatory quadrature. A sufficient condition for the unique existence of the rule is that the matrix S in (6) is non-singular. This implies that the basis functions can interpolate any function in the points $\mathbf{x}$. A Chebyshev set of functions on $[a, b]$, which can interpolate any function on any d points in $[a, b]$, is therefore suitable to construct general quadrature rules. Of those, polynomial basis functions are most popular. We denote by P_n the space of all polynomials up to degree n.

2.2 Definition of Newton-Cotes quadrature

A specific choice of points $\mathbf{x}$ is an equidistant set. Consider without loss of generality the interval $[a, b] = [-1, 1]$ and define the equidistant points[1]

$$\mathbf{x}_j^{NC} = -1 + 2\frac{j}{N-1}, \qquad j = 0, \ldots, N-1. \tag{9}$$

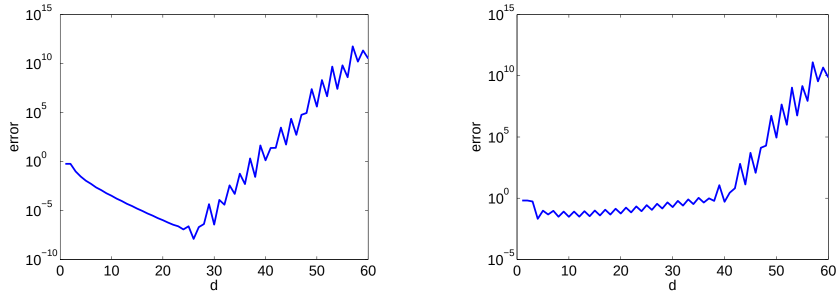
The interpolatory quadrature rule on $\mathbf{x}^{NC}$ that is exact for all polynomials in P_{d-1} for $w(x) = 1$ is called a Newton-Cotes rule. The two-point rule is the trapezoidal rule, with two constant weights and order 2. The three point rule is Simpson's rule, which has order 4.

2.3 Convergence of Newton-Cotes quadrature

Equispaced points appear in a variety of applications and Newton-Cotes quadrature therefore has great practical significance. Use of Newton-Cotes rules for high order is not recommended however, due to the numerical instability of such rules. Quadrature rules may be unstable if the weights are large and differ in sign. A common measure for the stability of quadrature rules is

$$\kappa(\mathbf{w}) = \sum_{i=1}^{N} |w_i|. \tag{10}$$

[1] Note that for numerical stability it is always a good idea to transform $[a, b]$ into $[-1, 1]$.



(a) $f(x) = \frac{1}{1+x^2}$ (b) $f(x) = \frac{1}{1+8x^2}$

Figure 2: The absolute error of Newton-Cotes quadrature as a function of the order d for two example functions with poles near the real axis. Both examples show divergence due to numerical instability.

If all weights are positive, we have $\kappa(\mathbf{w}) = \sum_{i=1}^N w_i = I[1]$. In all other cases $\kappa(\mathbf{w}) > I[1]$. The relevance of the quantity $\kappa(\mathbf{w})$ lies in the following basic bound on the worst case error. Assume that $\tilde{f}$ is a user-supplied function such that $|\tilde{f}(x) - f(x)| \leq \epsilon$. Then,

$$\begin{aligned} \left| \sum_{i=1}^N w_i \tilde{f}(x_i) - \sum_{i=1}^N w_i f(x_i) \right| &\leq \sum_{i=1}^N |w_i(\tilde{f}(x_i) - f(x_i))| \\ &\leq \sum_{i=1}^N |w_i| \epsilon = \epsilon \kappa(\mathbf{w}). \end{aligned}$$

This implies that, for example, round-off errors due to inexact arithmetic may be multiplied by the factor $\kappa(\mathbf{w})$. It is well-known that Newton-Cotes rules of high order (higher than 9 for $w(x) = 1$) have weights with mixed sign and round-off error may therefore affect convergence.

Worse still, even in exact arithmetic Newton-Cotes rules of high order may not converge (for $d \to \infty$) to the value of the integral [13]. This is due to the failure to converge of the underlying polynomial interpolation in equidistant points. A sufficient condition for convergence is that f is analytic in an ellipse centered at $(a+b)/2$, with a major axis of length $\frac{10}{8}(b-a)$ along the x-axis and a minor axis of length $\frac{6}{8}(b-a)$ into the complex plane [3].

We illustrate these properties in Figures 1 and 2. Fig. 1 displays $\kappa(\mathbf{w})$ for Newton-Cotes quadrature as a function of the order d. The quantity increases exponentially starting from $d = 9$. Fig. 2 shows the error of Newton-Cotes quadrature applied to the integrands $f(x) = \frac{1}{1+x^2}$ and $f(x) = \frac{1}{1+8x^2}$. In the former case (left panel), the process initially converges but then it diverges due to the numerical instability of the quadrature rules. In the latter case (right panel), the process does not converge because the function is not analytic in a sufficiently large region (the poles at $\pm\frac{i}{2\sqrt{2}}$ are too close to the real axis). After a while, divergence occurs again due to numerical instability. These examples were computed in Matlab in double precision.

3 Least-squares quadrature

We set out to construct stable high-order quadrature rules on equidistant point sets. In order to achieve this goal, we introduce redundancy by allowing the number of quadrature points N to be greater than the order d of the quadrature rule. We find the corresponding weights by solving a least-squares problem.

3.1 Formulation as a least-squares problem

For $N \geq d$, the counterpart of the linear system of equations (6) is given by

$$A\mathbf{w} = B, \tag{11}$$

where $A \in \mathbb{R}^{d \times N}$ is a rectangular matrix with elements

$$A_{i,j} = \phi_i(x_j), \qquad i = 1, \ldots, d, \quad j = 1, \ldots, N. \tag{12}$$

The right hand side vector B is the same as before, with elements given by (8).

The existence of a solution to (11) follows from the existence of interpolatory quadrature rules.

Lemma 3.1. *If $\{\phi_i\}_{i=1}^d$ is a Chebyshev set and $N \geq d$ then system* (11) *has at least one solution.*

Proof. One can construct an interpolatory quadrature rule of order d using d distinct arbitrary points by solving (6). The matrix S is non-singular because $\{\phi_i\}_{i=1}^d$ is a Chebyshev set. One can construct such a rule for any subset of d points from $\mathbf{x}$. Extending that rule with zero weights for the remaining points yields a solution to (11). □

It follows that the matrix A has rank d. The set W of all solutions to (11) is a linear space with dimension $N - d$ [11]. From this space, we choose the least squares solution $\mathbf{w}^*$, i.e., the one that minimizes $\|\mathbf{w}\|_2$:

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in W} \|\mathbf{w}\|_2. \tag{13}$$

The least squares solution $\mathbf{w}^*$ to (11) exists and is unique [11]. An explicit expression is found by solving the system

$$AA^T\mathbf{u} = B, \tag{14}$$

and setting

$$\mathbf{w}^* = A^T\mathbf{u}. \tag{15}$$

Note that $AA^T \in \mathbb{R}^{d \times d}$ is a square matrix. The set W consists of the sum of $\mathbf{w}^*$ and any vector in the kernel of A, i.e.,

$$W := \{\mathbf{w}^* + v \,|\, Av = 0\}. \tag{16}$$

In the following, we consider the case of quadrature rules that are exact for polynomials in P_{d-1}. There is considerable freedom in the choice of the basis functions ϕ_i. For interpolatory quadrature rules, this freedom is often used to reduce the condition number of matrix S in (6), for example by using Chebyshev polynomials. The least squares problem suggests a useful alternative.

3.2 Discrete orthogonal polynomials

Let us introduce the discrete scalar product associated with the vector $\mathbf{x}$,

$$u(f,g) = \sum_{j=1}^{N} f(x_j)g(x_j) \tag{17}$$

and the corresponding norm

$$\|f\|_u := \sqrt{u(f,f)}. \tag{18}$$

The scalar product $u(\cdot,\cdot)$ is positive definite on the space of polynomials up to degree $N-1$. Note that the polynomial p_N of degree N that vanishes at x_j satisfies $u(p_N, p_N) = 0$. It follows that we can define a finite sequence of discrete orthogonal polynomials. The n-th orthogonal polynomial p_n satisfies

$$p_n(x) \in P_n \qquad \text{and} \qquad \forall q \in P_{n-1} : u(p_n, q) = 0. \tag{19}$$

These polynomials are only defined up to a scaling factor. We define the normalized sequence of polynomials q_n by

$$q_n(x) = \frac{1}{\|p_n\|_u} p_n(x), \qquad n = 0, 1, \ldots, N-1. \tag{20}$$

3.3 Characterizing the least squares solution

Recall from (14) that the least squares problem is solved in terms of the matrix

$$C := AA^T. \tag{21}$$

One can verify that the elements of C are given explicitly by

$$C_{m,n} = \sum_{j=1}^{N} \phi_m(x_j)\phi_n(x_j) = u(\phi_m, \phi_n), \qquad m, n = 1, 2, \ldots, d. \tag{22}$$

It follows immediately that if the basis functions are orthogonal with respect to the discrete scalar product $u(\cdot,\cdot)$, then the matrix C is diagonal. If moreover the basis functions are orthonormal, the matrix C becomes the identity matrix $I_{d\times d}$. In that case, the solution of (14) is given by $u = B$ and we have

$$\mathbf{w}^* = A^T B. \tag{23}$$

The weights vector becomes a linear combination of the basis functions, evaluated in the quadrature points. Let $\Phi_i = [q_i(x_j)]_{j=1}^N$, then we obtain the following explicit expression for the weights:

$$\mathbf{w}^* = \sum_{i=0}^{d-1} \left(\int_a^b w(x) q_i(x)\,\mathrm{d}x \right) \Phi_i. \tag{24}$$

This expression holds for $N \geq d$ and for any arbitrary set of points $\mathbf{x}$. It is therefore valid for all interpolatory quadrature rules, where $N = d$, including rules with nonequispaced points.

Expression (24) is very informative. We may derive for example the asymptotic behaviour of the weights for fixed d as $N \to \infty$. Assume that the points $\mathbf{x}$ are equispaced on $[a,b] = [-1,1]$. Then the discrete inner product converges to the continuous L_2-inner product,

$$\lim_{N\to\infty} \frac{2}{N} u(f,g) = \lim_{N\to\infty} \frac{2}{N} f(x_j)g(x_j) = \int_{-1}^{1} f(x)g(x)\,\mathrm{d}x, \tag{25}$$

because the expression $\frac{2}{N}u(f,g)$ becomes a Riemann sum. It follows that the discrete orthogonal polynomials $q_i(x)$ converge to the classical Legendre polynomials $L_i(x)$, that are orthonormal on $[-1,1]$ with respect to the constant weight function. We have for example

$$q_0(x) = \frac{1}{\sqrt{N}} = q_0(1)L_0(x) \qquad \text{and} \qquad q_1(x) \to \frac{\sqrt{3}}{\sqrt{N}}x = q_1(1)L_1(x).$$

The normalizing factors in these expression appear because the classical Legendre polynomials satisfy $L_i(1) = 1$.

The elements of B, i.e., the moments (8), are now given explicitly by

$$B_i = \int_{-1}^{1} w(x)q_i(x)\,\mathrm{d}x \to q_i(1)\int_{-1}^{1} w(x)L_i(x)\,\mathrm{d}x. \tag{26}$$

These moments converge to (a scaling of) the coefficients of the expansion of $w(x)$ in Legendre polynomials. In the particular case where $w(x) = 1$, we have

$$\int_{-1}^{1} L_0(x)\,\mathrm{d}x = 2, \qquad \text{and} \qquad \int_{-1}^{1} L_i(x)\,\mathrm{d}x = 0, \qquad i = 2,3,\ldots.$$

Only the first moment is nonzero. From (24) it follows that the least squares weights converge to B_1 times the first row of A^T. This is a constant value,

$$\mathbf{w}^* \to \frac{2}{N}, \qquad N \to \infty. \tag{27}$$

This shows among other things that, for fixed d and sufficiently large N, the weights will become positive. The convergence is rather slow however, due to the slow convergence of the limit in (25). Up to a scaling, the discrete inner product is essentially an order 1 quadrature rule for the continuous inner product. The convergence can be increased by creating inner products that converge faster. By altering the inner product we can also study the case of general weight functions.

3.4 More general inner products

Consider the inner product

$$v(f,g) = \sum_{j=1}^{N} r_j f(x_j)g(x_j), \tag{28}$$

with values $r_j > 0$. The associated norm is $\|f\|_v = \sqrt{v(f,f)}$. We may find the weight vector $\mathbf{w}^*$ that minimizes $\|\mathbf{w}\|_V$ by solving the least squares problem

$$AR\,\tilde{\mathbf{w}} = B, \tag{29}$$

where $R \in \mathbb{R}^{N \times N}$ is a diagonal matrix with entries $R_{jj} = \sqrt{r_j}$. The least squares solution of (29) is found as

$$\tilde{\mathbf{w}^*} = (AR)^T \tilde{\mathbf{u}}$$

where $\tilde{u}$ is the unique solution of

$$AR(AR)^T \tilde{u} = B.$$

We obtain $\mathbf{w}^*$ as (see, for example, [11])

$$\mathbf{w}^* = R\tilde{\mathbf{w}^*}.$$

If the basis functions are orthonormal polynomials with respect to v, then the matrix $ARR^T A^T$ is diagonal and $\tilde{u} = B$. We find in this case that the weights are given explicitly by

$$\mathbf{w}^* = (AR)^T B, \tag{30}$$

or

$$\mathbf{w}^* = \sum_{i=0}^{d-1} \left(\int_a^b w(x) q_i(x) \, \mathrm{d}x \right) \Phi_i \cdot \mathbf{r},$$

where by $\Phi_i \cdot \mathbf{r}$ we mean the product elementwise. Compare this to (24).

We will now choose the coefficients r_i such that the discrete orthogonal polynomials converge to the polynomials that are orthogonal with respect to the weight function $w(x)$. For this, it is sufficient that

$$\lim_{N\to\infty} v(f,g) = \lim_{N\to\infty} \sum_{j=1}^{N} r_j f(x_j) g(x_j) = \int_a^b w(x) f(x) g(x) \, \mathrm{d}x.$$

This is possible if $w(x) \geq 0$ is a positive weight function. Any quadrature scheme with positive coefficients suits the pattern. We may for example use a composite trapezoidal rule that incorporates the weight function $w(x)$.

The first orthonormal discrete polynomial is given in general by

$$q_0(x) = \frac{1}{\sqrt{\sum_{j=1}^{N} r_j}}.$$

It follows that the first moment is given by

$$B_1 = \int_a^b w(x) q_0(x) \, \mathrm{d}x = \frac{1}{\sqrt{\sum_{j=1}^{N} r_j}} \int_a^b w(x) \, \mathrm{d}x \to \sqrt{\sum_{j=1}^{N} r_j}, \qquad N \to \infty,$$

because the sum of r_j converges to $\int_a^b w(x) \, \mathrm{d}x$ by construction. We have that $B_i \to 0$, $i = 1, 2, \ldots, d-1$. From (30) it follows that the weights in this case converge to B_1 times the first row of $(AR)^T$, and we find that

$$\mathbf{w}^* \to \mathbf{r}. \tag{31}$$

The least squares quadrature corresponding to the inner product simply converges to the vector $\mathbf{r} = [r_j]^T$, for fixed d and in the limit $N \to \infty$. Since $r_j > 0$, the weights of the least squares quadrature rule will become positive for sufficiently large N.

3.5 High-order corrections

From the results in the previous section §3.4, we see that the least squares quadrature rules can be interpreted as high-order corrections to low-order composite quadrature schemes. Assume that $[\mathbf{x}, \mathbf{r}^{[N]}]$ is a quadrature scheme for each N, with positive weights $r_j^{[N]} > 0$ and such that the quadrature approximation converges to $I[f]$ for $N \to \infty$. Compute the least squares quadrature rule $[\mathbf{x}, \mathbf{w}^*]$ that minimizes $\|\mathbf{w}\|_v$. This involves constructing polynomials that are orthogonal with respect to a weighted l_2 inner product (28), with $r_j^{[N]}$ as weights. Then $[\mathbf{x}, \mathbf{w}^*]$ has order d, regardless of the order of $[\mathbf{x}, \mathbf{r}^{[N]}]$, but the weights $\mathbf{w}^*$ converge to $\mathbf{r}^{[N]}$. Thus the difference $\mathbf{w}^* - \mathbf{r}^{[N]}$ becomes small. The rate of convergence depends on the order of $[\mathbf{x}, \mathbf{r}^{[N]}]$.

High-order corrections are well-known for the trapezoidal rule, based on the Euler-Maclaurin formula (see, for example [1] and references therein). These are sometimes called Gregory rules. The focus lies on constructing corrections near the endpoints only. This compares favorably to the current setting, where the corrections are spread over all weights, including those in the interior of the integration interval. On the other hand, the current construction guarantees positive weights, arbitrarily high order, equispaced points and fast construction.

4 An efficient construction algorithm

A stable algorithm to compute both interpolatory and least squares quadrature rules, on equidistant and arbitrary point sets, hinges on efficient methods to work with orthogonal polynomials. In this section we recall the use of the three-term recurrence formula to obtain a fast and stable method. This approach was first explored by Forsythe in [8].

4.1 Construction of the orthogonal polynomials

The polynomials p_n that are orthogonal with respect to the discrete inner product $v(\cdot, \cdot)$, as defined by (28), satisfy the well-known three term recurrence relation [10]

$$p_n(x) = \lambda_n(x - \alpha_n)p_{n-1}(x) - \lambda_n\beta_n p_{n-2}(x), \qquad n = 2, 3, \ldots, N-1. \tag{32}$$

with the coefficients α_n and β_n given by

$$\alpha_n = \frac{v(xp_{n-1}, p_{n-1})}{v(p_{n-1}, p_{n-1})} \tag{33}$$

and

$$\beta_n = \frac{v(xp_{n-1}, p_{n-2})}{v(p_{n-2}, p_{n-2})}. \tag{34}$$

The λ_n is a scaling parameter that can be used to normalize according to (20). Alternatively, one can choose $\lambda_n = 1$, which results in a monic polynomial.

The three-term recurrence relation is a useful and efficient way to construct the discrete orthogonal polynomial sequence. The recursion is started by setting

$$p_{-1}(x) \equiv 0 \qquad \text{and} \qquad p_0(x) = 1.$$

The recurrence relation is also a numerically stable way to evaluate these polynomials in any required point x. It takes $\mathcal{O}(N)$ operations to evaluate α_n and β_n for each value of n. Since n ranges from 1 to $d-1$ in our application, the orthogonal polynomials can be constructed in $\mathcal{O}(Nd)$ operations. If the recurrence coefficients are stored, the orthogonal polynomials can subsequently be evaluated in M points in $\mathcal{O}(Md)$ operations.

4.2 Computation of the moments

The right hand side of the linear system (29) requires the computation of the moments

$$\int_a^b w(x) q_i(x)\, \mathrm{d}x, \qquad i = 0, \ldots, d-1, \tag{35}$$

where $q_i(x)$ are the normalized discrete orthogonal polynomials. The computation of moments is a recurring problem in most methods for constructing quadrature rules. As such, it has been thoroughly investigated and described in the literature. We refer the reader to [9] for an account of the theory and methods.

In our numerical implementation, we have evaluated these moments simply by using known Gaussian quadrature rules for the weight function $w(x)$. This is a feasible approach for standard weight functions. We used a Gaussian rule of order d with $\lfloor \frac{d}{2} \rfloor$ points, which leads to $\lfloor \frac{d}{2} \rfloor$ evaluations per integrand. Each integrand evaluation requires $\mathcal{O}(d)$ computations using the recursive scheme from §4.1. This results in $\mathcal{O}(d^2)$ computational complexity.

4.3 Computational complexity

An efficient algorithm for constructing our least squares quadrature rules can be summarized as follows.

1. Construct the sequence of discrete orthogonal polynomials $p_n(x)$, $n = 0, \ldots, d-1$, by computing α_n and β_n for $n = 1, \ldots, d-1$. This step requires $\mathcal{O}(Nd)$ operations.

2. Normalize the sequence by computing the norms $\|p_n\|_u$ for $n = 0, \ldots, d-1$. This step also takes $\mathcal{O}(Nd)$ operations.

3. Compute the moments (35) of the orthogonal polynomials q_n. The computational complexity depends on the method that is used, but it is always independent of N. Our implementation requires $\mathcal{O}(d^2)$ operations.

4. Evaluate the weights by $\mathbf{w}^* = A^T R^T B$. This is another $\mathcal{O}(Nd)$ step.

Overall, the algorithm works in

$$O(Nd) + O(d^2) \tag{36}$$

steps. For a fixed d, this method is only linear in N. For interpolatory rules, where $d = N$, the method requires $O(d^2)$ operations.

5 Nonnegative least squares

The least squares rules are not optimal from the point of view of Tchakaloff's upper bound on the number of points required to obtain positive weights. The minimization of a (weighted) l_2-norm has the effect of spreading out the weights over all quadrature points – none of the weights are zero. It is known however that a rule with only d nonzero weights should exist on an equidistant grid with N points, provided that N is sufficiently large [4]. Such rules can be found by solving a least squares problem subject to inequality constraints, for which standard algorithms are readily available.

5.1 The NNLS formulation

A least squares problem with inequality constraints demanding a nonnegative solution is called a NNLS (nonnegative least squares) problem in [12]. A NNLS problem is defined as follows.

$$\text{Minimize } \|A\mathbf{x} - b\| \text{ subject to } x_j \geq 0,\ j = 1, \ldots, N. \tag{37}$$

In our setting, we are interested in the minimization of $\|A\mathbf{w} - b\|$ for the standard l_2 inner product and in the minimization of $\|AR\tilde{\mathbf{w}} - b\|$ for weighted inner products. Both cases fit the standard NNLS problem. Note that if $\tilde{w}_j \geq 0$, then $w_j = r_j \tilde{w}_j \geq 0$.

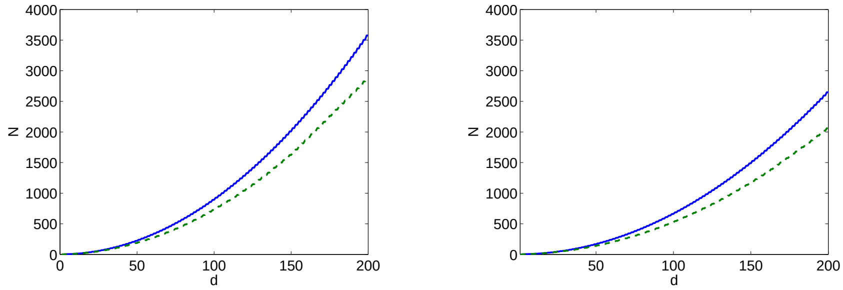
Problem (37) always has a solution. For an underdetermined matrix A, however, there may be infinitely many solutions. In that case, a solution is sought that maximizes sparsity of $\mathbf{x}$, i.e., one wishes to find the vector $\mathbf{x}$ with as much zero entries as possible.

5.2 Application to quadrature

In our setting, the solution of (37) always corresponds to a quadrature rule with positive weights. It is not necessarily the case however that $\|A\mathbf{w} - b\| = 0$, because a positive rule with the required exactness conditions may not exist. If N is sufficiently large, such that a quadrature rule with positive weights exists that satisfies $A\mathbf{w} = b$, then the solution to the NNLS problem will also be a quadrature rule with positive weights that satisfies $A\mathbf{w} = b$. Moreover, this rule has maximal sparsity. From theoretical considerations, we then know that the rule only has d nonzero entries. Note that this rule is not necessarily unique.

If $0 < \|A\mathbf{w} - b\| < \epsilon$, then the quadrature rule is not exact. The rule may have any number of nonzero weights, but none of the weights are negative. The rule is not necessarily exact for any polynomial. However, if $\epsilon \approx 0$, then it is reasonable to expect the quadrature approximation to have small error.

An algorithm to solve problem (37) exists that always converges. We refer the reader to [12, Ch.23] for a detailed description. It is an iterative algorithm, and convergence is proved by showing that the residual $\|A\mathbf{x} - b\|$ decreases in each iteration and that because of this the number of iterations is finite. In our numerical experiments, we used a Matlab implementation of the NNLS algorithm (`lsqnonneg`).



(a) $w(x) = 1$ (b) $w(x) = \sqrt{1 - x^2}$

Figure 3: Plot of the minimal number of equidistant points N required to construct a rule of order d with all weights positive for least squares quadrature (solid line) and nonnegative least squares quadrature (dashed line).

6 Numerical results

6.1 The minimal grid size: N versus d

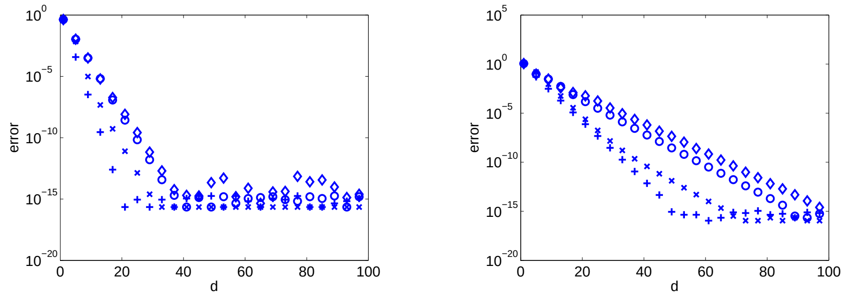
It was shown by Wilson in [18] that N grows proportional to d^2 for an equidistant grid. Fig. 3 shows the minimal number of equidistant points N that are required to obtain a quadrature rule of order d with all weights positive using the least squares algorithms of §4 and §5. The rules are constructed for the weight functions $w(x) = 1$ and $w(x) = \sqrt{1 - x^2}$ on the integration interval $[-1, 1]$. Rules with positive weights indeed exist for all orders and the relation $N \sim Cd^2$ can be confirmed numerically. For the constant weight function $w(x) = 1$ we find that $C \approx 0.09$ for least squares and $C \approx 0.07$ for nonnegative least squares. The computations involving the unconstrained least squares problem are most efficient. A rule of order 20 requires only $N = 36$. The rule of order 200 requires $N = 3.576$.[2] For the weight function $w(x) = \sqrt{1 - x^2}$, we find that the constants are smaller: $C \approx 0.07$ for the unconstrained and $C \approx 0.05$ for the constrained least squares rules.

Note that the values found by the NNLS algorithm correspond exactly to the minimal value of N such that the equidistant grid supports a quadrature rule of order d with positive weights. The unconstrained least squares problem however yields an acceptable approximation to this lower bound for both weight functions. Moreover, the corresponding algorithm is faster and entirely deterministic.

6.2 Convergence for increasing order d

We illustrated the numerical instability of classical Newton-Cotes quadrature in Fig. 2 in §2. Here, we repeat the experiment of Fig. 2 using least squares quadrature and nonnegative least squares quadrature. The results are shown in Fig. 4. Both types of least squares rules converge until machine precision is approximately reached. Once this precision is reached, it is also maintained at

[2] In our implementation, it takes 0.1 seconds to compute a rule of order $d = 200$ in $N = 3.576$ points (Matlab 7 running on an Intel Core Duo cpu at 2.4GHz).



(a) $f(x) = \frac{1}{1+x^2}$ (b) $f(x) = \frac{1}{1+8x^2}$

Figure 4: The absolute error as a function of the order d for two example functions for Gaussian quadrature (+), Clenshaw-Curtis (×), least squares (o) and nonnegative least squares (♢).

even higher order. This nicely illustrates the numerical stability of these high-order rules. Recall that the least squares rules require N points to achieve order d, whereas the nonnegative least squares rules only use function evaluations from d out of N possible equidistant points. The latter is therefore computationally much cheaper.

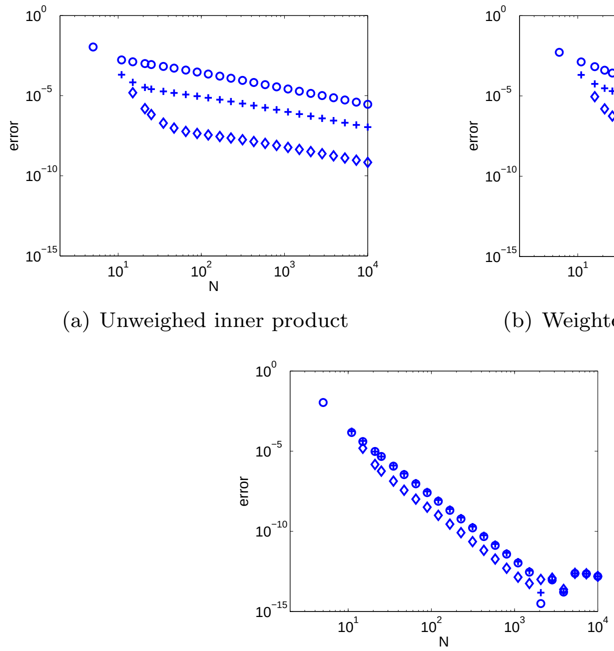
We also compared the results to Gaussian quadrature and Clenshaw-Curtis quadrature with the same number of quadrature points as nonnegative least squares rules. Both Gaussian quadrature and Clenshaw-Curtis quadrature converge faster. However, they require function values that are not equidistant. Note finally that Fig. 4(b) shows the typical kink phenomenon of Clenshaw-Curtis [16]. Clenshaw-Curtis starts out by converging as fast as Gaussian rules, until a kink occurs and the convergence rate slows down. The second rate of convergence is comparable to that of least squares quadrature.

6.3 Convergence for increasing grid size N

It was shown in §3.4 that, for a fixed d and increasing N, least squares quadrature rules converge to a composite quadrature rule of order 1. Perhaps surprisingly, this implies convergence as $N \to \infty$ while the order d remains fixed. It was further shown in §3.4 that, by choosing weighted inner products (28) rather than the standard l_2 inner product (17), the least squares quadrature rules can be made to converge to any composite quadrature scheme with positive weights.

In our next experiment we vary the weights $\mathbf{r}$ of the weighted inner product $v(\cdot,\cdot)$, as defined by (28). The results of §3.4 imply that the weights $\mathbf{w}$ of the corresponding least squares quadrature rule will converge to $\mathbf{r}$ as N becomes large. We chose $\mathbf{r}$ to be the unit vector, the weights of a composite trapezoidal rule and the weights a composite Simpson rule on an equidistant grid. The results are shown in Fig. 5. The figure confirms convergence for fixed order d and increasing N. The convergence rate is improved by choosing a higher order quadrature scheme as weights for the inner product. The rate of convergence as $N \to \infty$ remains the same as the order d is increased ($d = 5, 10, 15$ in the figure), but the accuracy improves with d.

Recall from §4.3 that the computational complexity of constructing these



(a) Unweighed inner product

(b) Weighted by trapezoidal rule

(c) Weighted by Simpson's rule

Figure 5: The absolute error as a function of N for $f(x) = \frac{1}{1+x^2}$. The figure shows the curve for $d = 5$ (o), $d = 10$ (+) and $d = 15$ ($\Diamond$).

rules is only linear in N. It is therefore relatively cheap to compute these quadrature rules.

6.4 Non-equidistant points

The methods of this paper also apply to a set $\mathbf{x}$ of non-equidistant spaced points. It was shown by Davis in [4, 5] that weights can be positive for sufficiently large N, as long as the distribution of the points $\mathbf{x}$ is dense in the integration interval $[a, b]$ for increasing N.

We repeat the experiment of Fig. 2 once more. For each order d, we generated random vectors of increasing size with entries in $[-1, 1]$, until a vector $\mathbf{x}$ was found that supported a least squares quadrature rule with positive weights of order d. The results are shown in Fig. 6. The left panel shows stable convergence for increasing order d. The right panel shows the number of quadrature points N as a function of d that were required to obtain a rule with positive weights of order d. No attempt was made to find rules of order d with smaller N. The existence of Gaussian quadrature shows that the minimal number of points required is only $N = \lceil \frac{d}{2} \rceil$. The values in Fig. 6(b) are of course much larger than this minimum.

We end the paper with the following experiment. Assume that a random vector of length N is given and that function samples are only available in this set. Now compute least squares quadrature rules of increasing order d and approximate the integral of the sampled function with these rules. The results

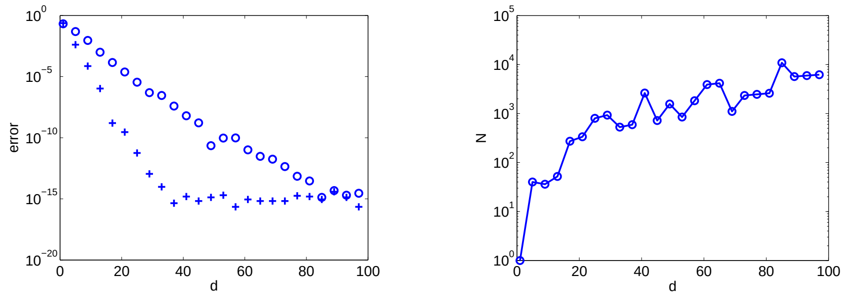
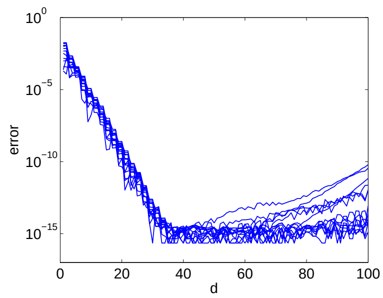
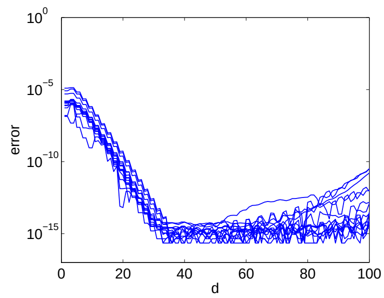


Figure 6: Illustration of quadrature rules with positive weights on a random set of points for increasing order d. The left panel shows the absolute error for $f(x) = \frac{1}{1+x^2}$ and $f(x) = \frac{1}{1+8x^2}$. The right panel shows the number of random quadrature points N as a function of d.



(a) Standard inner product



(b) Weighted by trapezoidal rule

Figure 7: The absolute error of least squares quadrature rules of increasing order d on 15 random vectors $\mathbf{x}$ of length $N = 1025$ for the function $f(x) = \frac{1}{1+x^2}$.

of this experiment are shown in Fig. 7 for 15 random vectors. We used the standard l_2 inner product in Fig. 7(a) and an inner product weighted by the composite trapezoidal rule in Fig. 7(b). The latter yields higher initial accuracy. Both figures exhibit convergence until the weights start having mixed sign, at which point the error curves can go up again. Note that the graphs for different random vectors $\mathbf{x}$ are in good agreement.

Finally, we recall that these results are equivalent to computing the least squares discrete approximation of the integrand in the sampling points.

7 Concluding remarks

We argued in the introduction that the least squares interpretation of quadrature formulae holds several advantages. The three points that were raised are illustrated in the numerical results. A useful stopping criterium for the least squares approximation in Fig. 7 is that the weights of the corresponding least squares quadrature rule should be positive. Divergence may be seen otherwise. The stability of the implementation is illustrated by the convergence to almost machine precision for all examples, with all computations including construction of the quadrature rules carried out in double precision without exact arithmetic of any kind. Efficiency moreoever is illustrated by the fact that the required computations for all figures in this paper combined were created in a matter of seconds, with the exception of Fig. 3 showing the minimal number of quadrature points N for nonnegative least squares quadrature rules as a function of the order. Finally, the unconventional use of low-order quadrature rules as weights for the inner product leads to improvements in the accuracy of the computed integral as shown in Fig. 5 and Fig. 7(b).

Acknowledgements

The author would like to thank Ronald Cools (K.U. Leuven, Belgium) for several interesting discussions on this topic.

References

[1] B. K. Alpert. Hybrid Gauss-trapezoidal quadrature rules. *SIAM J. Sci. Comput.*, 20(5):1551–1584, 1999.

[2] R. Cools. Constructing cubature formulae: The science behind the art. *Acta Numer.*, 6:1–54, 1997.

[3] P. J. Davis. On a problem in the theory of mechanical quadratures. *Pacific J. Math.*, 5:669–674, 1955.

[4] P. J. Davis. A construction of nonnegative approximate quadrature. *Math. Comp.*, 21:578–582, 1967.

[5] P. J. Davis. Approximate integration rules with nonnegative weights. In A. K. Aziz, editor, *Lectures in differential equations*, volume 2, pages 233–256, 1969.

[6] P. J. Davis and P. Rabinowitz. *Methods of numerical integration.* Computer Science and Applied Mathematics. Academic Press, New York, 1984.

[7] G. A. Evans. *Practical numerical integration.* John Wiley & Sons, Chichester, 1993.

[8] G. E. Forsythe. Generation and use of orthogonal polynomials for data-fitting with a digital computer. *J. SIAM*, 5(2):74–88, 1956.

[9] W. Gautschi. Moments in quadrature problems. *Comput. Math. Appl.*, 33:105–118, 1997.

[10] W. Gautschi. *Orthogonal polynomials: computation and approximation.* Clarendon Press, Oxford, 2004.

[11] G. H. Golub and C. F. van Loan. *Matrix Computations.* Johns Hopkins University Press, 3rd edition, 1996.

[12] C. L. Lawson and R. J. Hanson. *Solving least squares problems.* Classics in Applied Mathematics. SIAM, Philadelphia, 1996.

[13] G. Pólya. Über die Konvergenz von Quadarturverfahren. *Math. Z.*, 37:264–286, 1933.

[14] A. H. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice-Hall, Englewood Cliffs, N.J., 1971.

[15] V. Tchakaloff. Formules de cubature mécaniques à coefficients non négatifs. *Bull. Sci. Math.*, 81:123–134, 1957.

[16] L. N. Trefethen. Is Gauss quadrature better than Clenshaw-Curtis? *SIAM Rev.*, 50(1):67–87, 2008.

[17] M. W. Wilson. Discrete least squares and quadrature formulas. *Math. Comp.*, 24(110):271–282, 1970.

[18] M. W. Wilson. Necessary and sufficient conditions for equidistant quadrature formula. *SIAM J Numer. Anal.*, 4(1):134–141, 1970.