A geometrical interpretation of the addition of nodes to an interpolatory quadrature rule while preserving positive weights

van den Bos, L. M.M.; Sanderse, B.

DOI
10.1016/j.cam.2021.113430

Publication date
2021

Document Version
Final published version

Published in
Journal of Computational and Applied Mathematics

Journal of Computational and Applied Mathematics 391 (2021) 113430

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A geometrical interpretation of the addition of nodes to an interpolatory quadrature rule while preserving positive weights

L.M.M. van den Bos [a,b,*], B. Sanderse [a]

[a] Centrum Wiskunde & Informatica, P.O. Box 94079, 1090 GB, Amsterdam, The Netherlands
[b] Delft University of Technology, P.O. Box 5, 2600 AA, Delft, The Netherlands

ARTICLE INFO

Article history:
Received 12 July 2020
Received in revised form 8 December 2020

Keywords:
Quadrature rules
Numerical integration
Interpolation

ABSTRACT

A novel mathematical framework is derived for the addition of nodes to univariate and interpolatory quadrature rules. The framework is based on the geometrical interpretation of the Vandermonde matrix describing the relation between the nodes and the weights and can be used to determine all nodes that can be added to an interpolatory quadrature rule with positive weights such that the positive weights are preserved. In the case of addition of a single node, the derived inequalities that describe the regions where nodes can be added are explicit. Besides addition of nodes these inequalities also yield an algorithmic description of the replacement and removal of nodes. It is shown that it is not always possible to add a single node while preserving positive weights. On the other hand, addition of multiple nodes and preservation of positive weights is always possible, although the minimum number of nodes that need to be added can be as large as the number of nodes of the quadrature rule. In case of addition of multiple nodes the inequalities describing the regions where nodes can be added become implicit. It is shown that the well-known Patterson extension of quadrature rules is a special case that forms the boundary of these regions and various examples of the applicability of the framework are discussed. By exploiting the framework, two new sets of quadrature rules are proposed. Their performance is compared with the well-known Gaussian and Clenshaw–Curtis quadrature rules, demonstrating the advantages of our proposed nested quadrature rules with positive weights and fine granularity.

1. Introduction

This article is concerned with the addition of nodes to univariate and interpolatory quadrature rules with positive weights. If such a quadrature rule is given, the goal is to determine all sequences of nodes such that, upon adding all nodes from such a sequence to the rule, an interpolatory quadrature rule with positive weights is again obtained. The motivation of this problem is twofold. Firstly, approximations of integrals computed using interpolatory quadrature rules with positive weights converge for any absolute continuous function [1–3]. Secondly, nested quadrature rules allow for straightforward refinements of the quadrature rule approximation, which is especially relevant if the integrand is computationally expensive.

* Corresponding author at: Centrum Wiskunde & Informatica, P.O. Box 94079, 1090 GB, Amsterdam, The Netherlands.
E-mail address: l.m.m.van.den.bos@cwi.nl (L.M.M. van den Bos).

https://doi.org/10.1016/j.cam.2021.113430

Possibly the best-known interpolatory quadrature rule is the Gaussian quadrature rule [4], which exists for virtually any probability distribution with finite moments. It has positive weights and maximal polynomial degree. However, the nodes are not nested. The Gauss–Kronrod quadrature rule is an extension of a Gaussian quadrature rule, such that two nested rules with positive weights are obtained [5,6]. The Gauss–Kronrod–Patterson quadrature rule [7,8] further extends this idea by repeatedly applying the same algorithm, such that a sequence of nested rules is obtained. However, it does not exist for any distribution [9,10]. Even though many other extensions have been proposed over the years [11–13], in general it is difficult to obtain a series of nested quadrature rules with positive weights. Moreover often the smallest possible granularity between two consecutive nested quadrature rules can only be found by exhaustive search [14].

Another large group of well-known quadrature rules is formed by the Clenshaw–Curtis quadrature rules [15], or simply those quadrature rules that are based on Chebyshev approximations (the Clenshaw–Curtis rule is formed by the Chebyshev extrema). Besides having excellent interpolation properties [16], it is well-known that these quadrature rules have positive weights if the distribution under consideration is uniform (explicit expressions are known [17]). Moreover for non-uniform distributions, the condition number of the quadrature rule converges to unity [3]. However, the vanilla Clenshaw–Curtis nodes are only nested for exponentially growing numbers of nodes [18].

Both the Gaussian and Clenshaw–Curtis quadrature rules have explicitly predefined nodes based on the roots of orthogonal polynomials. This results in accurate quadrature rules, but the construction of an accurate nested quadrature rule with fine granularity based on these rules remains notoriously difficult.

In this article the goal is to propose a geometrical framework for the addition of nodes to an interpolatory quadrature rule with positive weights and use this framework to determine all interpolatory quadrature rules with positive weights that extend a rule based on predefined nodes. It will be demonstrated rigorously that the boundary of the set that contains all nodes that can be added is equivalent to the Patterson extension of quadrature rules, such that a special case of the framework is an extension of the aforementioned Gaussian quadrature rule families.

The approach taken is based on the geometrical interpretation of the linear system describing the nodes and the weights [19–21], which yields a necessary and sufficient condition for a quadrature rule to have positive weights. The framework embeds previous results on the removal of nodes from quadrature rules [19,22,23] and describes, besides a geometrical description of all nodes that can be added to a quadrature rule, algorithms that can be used to construct and modify interpolatory quadrature rules with positive weights.

The addition and replacement of a single node can be determined analytically, whereas numerical methods are required to determine the bounds on the sets describing multiple nodes. The focus of this article is mainly on the geometrical and mathematical aspects and not on the numerical aspects of the proposed algorithms. However, to illustrate the potential of the framework, two straightforward examples of quadrature rules with positive weights that can be constructed by exploiting the proposed techniques are discussed.

In Section 2 the nomenclature used in this article is discussed, including the motivation behind enforcing positive weights. In Section 3 the problem of adding a single node to a quadrature rule is considered, which can be solved analytically. It is not always possible to add a node to a quadrature rule such that the resulting rule has positive weights. Therefore the theory is extended to adding multiple nodes in Section 4, where the results developed for adding a single node will be used extensively. It is always possible to add multiple nodes to a quadrature rule, provided that any number of nodes may be added to the rule. To demonstrate the advantages of nested quadrature rules with positive weights, two quadrature rules that are derived in this work are compared with the well-known Gaussian and Clenshaw–Curtis quadrature rule. The details and results of this numerical experiment are discussed in Section 5. Conclusions and suggestions for future work are discussed in Section 6.

2. Preliminaries

The quadrature rule nomenclature relevant for this article is discussed in Section 2.1. The relevance of positive weights and the relation between positive weights and accuracy of a quadrature rule are briefly reviewed in Section 2.2. The mathematical notion of adding nodes to a quadrature rule can be interpreted as a non-trivial extension of the removal of nodes [19,22,23], which is briefly discussed in Section 2.3. Finally, the problem setting of this article and the main results obtained from this work are summarized mathematically in Section 2.4.

2.1. Nomenclature

A quadrature rule is a well-known approach to approximate a weighted integral in the interval $\Omega = [a, b] \subset \mathbb{R}$ with $-\infty \leq a < b \leq \infty$. The weighting function is a positive density function $\rho\colon \Omega \to [0, \infty)$. The main interest is to approximate the integral over a given continuous function $u\colon \Omega \to \mathbb{R}$, i.e. to approximate the following operator:

$$\mathcal{I}u = \int_{\Omega} u(x)\,\rho(x)\,\mathrm{d}x = \int_{a}^{b} u(x)\,\rho(x)\,\mathrm{d}x.$$

A quadrature rule approximates this integral by means of a weighted average, consisting of nodes and weights, which we denote by $X_N = \{x_0, \ldots, x_N\} \subset \Omega$ and $W_N = \{w_0, \ldots, w_N\} \subset \mathbb{R}$ respectively. The quadrature rule is the following

operator $\mathcal{A}_N$:

$$\mathcal{A}_N u := \sum_{k=0}^{N} u(x_k) w_k \approx \mathcal{I} u.$$

It is common to measure the consistency of this construction by means of polynomial degree. The polynomial degree of a quadrature rule is defined as the maximum polynomial degree the quadrature rule integrates exactly, or equivalently: a quadrature rule of degree K has the property

$$\mathcal{A}_N \varphi = \mathcal{I} \varphi, \text{ for all } \varphi \in \mathbb{P}(K), \tag{2.1}$$

where $\mathbb{P}(K)$ denotes the space of all univariate polynomials of degree K or less. This definition is only meaningful if ρ has finite moments, so that is assumed to be the case throughout this article.

A quadrature rule is called *interpolatory* if the dimension of $\mathbb{P}(K)$ is larger than or equal to the number of nodes, or in other words, if $K \geq N$. Such quadrature rules can be formed by integrating the polynomial interpolant of u using the nodes X_N. As the title of this article suggests, these quadrature rules are the main focus of this work: throughout this article the interest is mainly in rules with $K = N$ (though quadrature rules with $K > N$, such as the Gaussian quadrature rules, will also be considered).

The operators $\mathcal{A}_N$ and $\mathcal{I}$ and the space $\mathbb{P}(K)$ are linear, so if $K = N$, (2.1) defines a linear system that can be used to determine the weights, given the nodes and the moments of the distribution. Throughout this article a monomial basis of $\mathbb{P}(K)$ is considered. In this case, the matrix of the linear system is the well-known Vandermonde matrix, denoted as follows:

$$\underbrace{\begin{pmatrix} x_0^0 & \cdots & x_N^0 \\ \vdots & \ddots & \vdots \\ x_0^N & \cdots & x_N^N \end{pmatrix}}_{V(X_N)} \begin{pmatrix} w_0 \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} \mu_0 \\ \vdots \\ \mu_N \end{pmatrix}, \tag{2.2}$$

with μ_k the raw moments of ρ:

$$\mu_k = \int_\Omega x^k \rho(x) \, \mathrm{d}x.$$

Throughout this article it is assumed that μ_k is known exactly for all k. The notation $V(X_N)$ is used for the matrix of this linear system. It is well-known that

$$\det V(X_N) = \prod_{0 \leq i < j \leq N} (x_j - x_i), \tag{2.3}$$

such that, given the nodes, (2.2) defines a unique solution of the weights provided that all nodes are distinct.

2.2. Accuracy of quadrature rules

In this article the focus is on constructing interpolatory quadrature rules with non-negative weights (which we will call with a little abuse of nomenclature a positive quadrature rule). An approximation of an integral by means of such a quadrature rule converges if the integrand is sufficiently smooth [1], which can among others be demonstrated by applying the Lebesgue inequality [3], provided that Ω is bounded. To this end, let u be given and let φ_N be the best approximation polynomial [24] of degree N of u, i.e. $\varphi_N = \arg\min_{\varphi \in \mathbb{P}(N)} \|\varphi - u\|_\infty$. Then

$$\begin{aligned} |\mathcal{A}_N u - \mathcal{I} u| &\leq (\|\mathcal{A}_N\|_\infty + \|\mathcal{I}\|_\infty) \|u - \varphi_N\|_\infty \\ &= (\|\mathcal{A}_N\|_\infty + \mu_0) \|u - \varphi_N\|_\infty. \end{aligned}$$

Here, it holds that

$$\|\mathcal{A}_N\|_\infty = \sup_{\|u\|_\infty = 1} |\mathcal{A}_N u| = \sum_{k=0}^{N} |w_k| = \sum_{k=0}^{N} w_k = \mu_0,$$

where it is used that $|w_k| = w_k$. Hence the following inequality is obtained:

$$|\mathcal{A}_N u - \mathcal{I} u| \leq 2 \, \mu_0 \inf_{\varphi \in \mathbb{P}(N)} \|u - \varphi\|_\infty. \tag{2.4}$$

This shows many similarities with the classical Lebesgue inequality [16,24] and demonstrates that if u can be approximated well using a polynomial, it can be integrated using a quadrature rule with positive weights. Similar results exist for unbounded Ω [3,25].

Two well-known interpolatory quadrature rules with positive weights are the Clenshaw–Curtis and Gaussian quadrature rules.

The Clenshaw–Curtis quadrature rule [15] has nodes X_N that are defined as follows for $\Omega = [-1, 1]$:

$$x_k = \cos\left(\frac{k}{N}\pi\right), \text{ for } k = 0, \dots, N. \tag{2.5}$$

The Clenshaw–Curtis quadrature rule has positive weights if the uniform distribution is considered and for any other distribution with bounded support the sum of the absolute weights becomes arbitrary close to μ_0 for large N [3]. The quadrature rule is nested for specific levels: it holds that $X_{N_L} \subset X_{N_{L+1}}$ with $N_L = 2^L$ (for $L = 1, 2, \dots$).

The nodes of the Gaussian quadrature rule [4] are defined as the roots of the orthogonal polynomials with respect to the distribution ρ under consideration, e.g. Legendre polynomials for the uniform distribution, Jacobi polynomials for the Beta distribution, etc. The uniquely defined rules always have positive weights and with $N + 1$ nodes the rule has degree $2N + 1$, however the rules are not nested.

The Gauss–Kronrod and Gauss–Patterson quadrature rules are extensions of Gaussian quadrature rules such that upon adding M nodes (with $M = N + 2$ for the Gauss–Kronrod rule) to a rule of $N + 1$ nodes, a (not necessarily positive) rule of degree $N + 2M$ is obtained [5,8]. The Patterson extension is also applicable to non-Gaussian quadrature rules, though possibly complex-valued nodes can be obtained. The idea is to solve the following problem for $x_{N+1}, \dots, x_{N+M}$, given quadrature rule nodes X_N:

$$\int_\Omega x^j \left[\prod_{k=0}^{N+M}(x - x_k)\right] \rho(x)\,\mathrm{d}x = 0, \text{ for } j = 0, \dots, M - 1. \tag{2.6}$$

Then the obtained rule has degree $N+2M$ [3, Theorem 5.1.3], is defined uniquely, and possibly has complex-valued nodes. By construction, a Gaussian quadrature rule is obtained if $M = N+1$ (the weights of the nodes in X_N become zero). These rules are reobtained as a special case in the framework discussed in this work.

2.3. Removal of nodes

The primary focus of this article is on the *addition* of nodes, but the obtained mathematical expressions can be interpreted as reversing the *removal* of nodes from an existing quadrature rule. Using Carathéodory's theorem, it can be shown that for each positive interpolatory quadrature rule X_N, W_N there exist two nodes x_{k_0} and x_{k_1} such that $X_N \setminus \{x_{k_0}\}$ and $X_N \setminus \{x_{k_1}\}$ both form the nodes of interpolatory quadrature rules with positive weights [19,22,23,26]. The details are discussed in the constructive proof of the following theorem.

Theorem 1 (*Carathéodory's Theorem*). *Let $\mathbf{v}_0, \dots, \mathbf{v}_N$ be $N + 1$ vectors spanning an N-dimensional space V. Let $\mathbf{v} \in V$ be such that $\mathbf{v} = \sum_{k=0}^{N} a_k \mathbf{v}_k$ with all $a_k \geq 0$. Then there exist non-negative b_k and a $k_0 \in \{0, \dots, N\}$ such that*

$$\mathbf{v} = \sum_{\substack{k=0 \\ k\neq k_0}}^{N} b_k \mathbf{v}_k.$$

Proof. The vectors $\mathbf{v}_0, \dots, \mathbf{v}_N$ are linearly dependent, since these are $N + 1$ vectors spanning an N-dimensional space. Hence there exists a vector $\mathbf{c} = (c_0, \dots, c_N)^{\mathrm{T}} \neq \mathbf{0}$ such that

$$\sum_{k=0}^{N} c_k \mathbf{v}_k = \mathbf{0}.$$

Hence for any $\alpha \in \mathbb{R}$, we have that

$$\mathbf{v} = \sum_{k=0}^{N} (a_k - \alpha c_k)\mathbf{v}_k.$$

In particular, consider the following α and k_0:

$$\alpha = \min\left(\frac{a_k}{c_k} \mid c_k > 0\right) =: \frac{a_{k_0}}{c_{k_0}}.$$

With these choices it holds that $a_k - \alpha c_k \geq 0$ for all k and $a_{k_0} - \alpha c_{k_0} = 0$, concluding the proof as follows:

$$\mathbf{v} = \sum_{\substack{k=0 \\ k\neq k_0}}^{N} (a_k - \alpha c_k)\mathbf{v}_k. \quad \square$$

The theorem can be used straightforwardly to remove nodes from a quadrature rule. To this end, let the positive interpolatory quadrature rule X_N and W_N be given. The goal is to construct an interpolatory quadrature rule using N nodes from X_N (which consists of $N+1$ nodes). Therefore, let $\mathbf{v}_0, \dots, \mathbf{v}_N$ be the columns of the Vandermonde matrix of degree $N-1$, i.e. $\mathbf{v}_k = (x_k^0, \dots, x_k^{N-1})^{\mathrm{T}}$. Then $\mathbf{v}_k$ are $N+1$ vectors spanning an N-dimensional space. The proof of Carathéodory's theorem yields that there exist a vector $\mathbf{c}$, scalar α, and index k_0 such that

$$\mu_j = \sum_{\substack{k=0 \\ k\neq k_0}}^{N} x_k^j (w_k - \alpha c_k), \text{ for all } j = 0, \dots, N-1.$$

Moreover, we have that $w_{k_0} - \alpha c_{k_0} = 0$, so by using $X_{N-1} = \{x_k \in X_N \mid k \neq k_0\}$ and $W_{N-1} = \{w_k - \alpha c_k \mid k \neq k_0\}$ a positive interpolatory quadrature rule is obtained. Notice that the vector $\mathbf{c}$ is computable, since it is a null vector of the Vandermonde matrix of degree $N-1$ (which is an $N \times (N+1)$-matrix).

This approach can be used to compute nested quadrature rules, but limits the accuracy of those quadrature rules to the initial rule of which nodes are removed. It is of less use if this rule is inadequately accurate or if no such rule is available. A possible approach to alleviate this is to use random samples as initial quadrature rule [27], though such samples do not accurately integrate higher order moments. The necessity of an existing quadrature rule is one of the main motivations to consider the *addition* of nodes, since that does not require the computation of an initial quadrature rule of sufficient accuracy.

2.4. Problem setting and main results

The problem studied in this article is how to add nodes to a positive interpolatory quadrature rule such that it remains positive and interpolatory. To formulate this mathematically, let a positive interpolatory quadrature rule X_N, W_N be given. Then the goal is to determine, for given M, all nodes such that the set X_{N+M} contains the nodes of a positive interpolatory quadrature rule and such that the rules are nested, i.e. $X_N \subset X_{N+M}$. To keep the nomenclature concise, we will refer to this problem as adding M nodes to a positive interpolatory quadrature rule, where by "adding" we always mean addition such that the resulting quadrature rule has positive weights. Moreover the number of nodes added to a quadrature rule should be minimal, so we are also interested in the minimal value of M (with $M > 0$) such that a positive interpolatory quadrature rule with nodal set X_{N+M} exists.

If a positive quadrature rule is given that is not interpolatory, i.e. a quadrature rule such that $\mathcal{A}_N\varphi = \mathcal{I}\varphi$ for all $\varphi \in \mathbb{P}(K)$ with $K < N$, a positive interpolatory quadrature rule can be deduced from this rule by repeatedly applying Theorem 1. Therefore we assume in this article without loss of generality that all quadrature rules are interpolatory.

The approach is to formulate, for given M, a necessary and sufficient condition for all M nodes that can be added. This condition can be used firstly to determine whether such nodes exist for a specific M and secondly to determine the nodes themselves. Moreover the derived theory allows for specific adjustments of quadrature rules. These adjustments consist of replacing and removing nodes from the quadrature rule, in such a way that the degree of the rule is not affected.

The analysis is split into two sections. The addition of a single node ($M = 1$) can be solved analytically and is discussed in Section 3. The addition of multiple nodes ($M > 1$) can only be done analytically for special cases. Based on the theory for $M = 1$, this problem is analyzed in Section 4.

3. Addition of one node

Let X_N, W_N be a positive interpolatory quadrature rule. The goal is to determine all x_{N+1} such that $X_{N+1} = X_N \cup \{x_{N+1}\}$ forms the nodal set of a positive interpolatory quadrature rule, i.e. there exists a set of non-negative weights W_{N+1} such that

$$\sum_{k=0}^{N+1} x_k^j w_k^{(N+1)} = \mu_j, \text{ for } j = 0, \dots, N+1.$$

Here, $w_k^{(N+1)}$ are the weights in the set W_{N+1} and μ_j is assumed to be known. Notice that in general W_N and W_{N+1} will completely differ, so we use the following notation for any N:

$$W_N = \{w_0^{(N)}, \dots, w_N^{(N)}\}.$$

Moreover, with a little abuse of notation we will use $w_k^{(N)} = 0$ for all $k > N$.

In Section 3.1 we derive a necessary and sufficient condition for such an x_{N+1} to exist, which depends on the current nodes, weights, and moment μ_{N+1}. As such, the developed theory provides practical adjustments of a quadrature rule. These constitute addition and replacement of a node, without reducing the degree of the interpolatory quadrature rule. The details are discussed in Section 3.2 and will be very useful in the remainder of this article. In Section 3.3 the Patterson extension is discussed in light of the derived adjustments and some basic applications of the derived procedures are discussed, including the construction of a (partially) nested quadrature rule with positive weights.

3.1. Positive weight criterion

The key notion is that if the node x_{N+1} is given, a vector $\mathbf{c} = (c_0, \ldots, c_{N+1})^{\mathrm{T}}$ can be constructed such that $w_k^{(N+1)} = w_k^{(N)} + c_k$ (for $k = 0, \ldots, N+1$). This is the vector used in Section 2.3 to remove nodes from a rule. If this vector is such that $c_k \geq -w_k^{(N)}$, then $w_k^{(N+1)} \geq 0$, which is the primary goal. In this section, these properties are translated to conditions on x_{N+1} that describe in which cases a node can be added to a quadrature rule.

The interpolatory quadrature rule X_N, W_N has degree N, so after adding x_{N+1} the following should hold to ensure that the new rule is interpolatory:

$$\mu_j = \sum_{k=0}^{N} x_k^j w_k^{(N)} = \sum_{k=0}^{N+1} x_k^j w_k^{(N+1)}, \quad \text{for } j = 0, \ldots, N.$$

From this, it follows for $j = 0, \ldots, N$ that (using $w_{N+1}^{(N)} = 0$):

$$0 = \sum_{k=0}^{N+1} x_k^j w_k^{(N+1)} - \sum_{k=0}^{N+1} x_k^j w_k^{(N)} = \left(\sum_{k=0}^{N+1} x_k^j w_k^{(N)} + \sum_{k=0}^{N+1} x_k^j c_k \right) - \sum_{k=0}^{N+1} x_k^j w_k^{(N)} = \sum_{k=0}^{N+1} x_k^j c_k. \tag{3.1}$$

The goal is to construct X_{N+1} and W_{N+1} such that they form a quadrature rule of degree $N+1$. Hence with $\mu_{N+1} = \int_\Omega x^{N+1} \rho(x)\,\mathrm{d}x$ given, it should hold that

$$\sum_{k=0}^{N+1} x_k^{N+1} w_k^{(N+1)} = \mu_{N+1},$$

which can be expressed in terms of the vector $\mathbf{c}$ as

$$\varepsilon_{N+1} \coloneqq \mu_{N+1} - \sum_{k=0}^{N} x_k^{N+1} w_k^{(N)} = \sum_{k=0}^{N+1} x_k^{N+1} c_k. \tag{3.2}$$

The value of ε_{N+1} can be interpreted as the approximation error of the quadrature rule with nodes X_N and weights W_N with respect to μ_{N+1}. Combining (3.1) and (3.2) yields the following system of linear equations for the vector $\mathbf{c}$:

$$\begin{pmatrix} x_0^0 & \cdots & x_N^0 & x_{N+1}^0 \\ \vdots & \ddots & \vdots & \vdots \\ x_0^N & \cdots & x_N^N & x_{N+1}^N \\ x_0^{N+1} & \cdots & x_N^{N+1} & x_{N+1}^{N+1} \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_N \\ c_{N+1} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \varepsilon_{N+1} \end{pmatrix},$$

or more compactly:

$$V(X_{N+1})\,\mathbf{c} = \boldsymbol{\varepsilon},$$

with $\boldsymbol{\varepsilon} = (0, \ldots, 0, \varepsilon_{N+1})^{\mathrm{T}}$. The vector $\boldsymbol{\varepsilon}$ has a large number of zeros so it is convenient to apply Cramer's rule to this linear system, which yields

$$c_k = \frac{\det V_k(X_{N+1})}{\det V(X_{N+1})}, \tag{3.3}$$

where $V_k(X_{N+1})$ is equal to $V(X_{N+1})$ with the kth column replaced by $\boldsymbol{\varepsilon}$, where the indexing of columns is started with 0. This expression can be simplified by noticing that

$$\det V_k(X_{N+1}) = (-1)^{(N+2)+(k+1)} \varepsilon_{N+1} \det V(X_{N+1} \setminus \{x_k\}) = (-1)^{N+k+1} \varepsilon_{N+1} \det V(X_{N+1} \setminus \{x_k\}),$$

with $V(X_{N+1} \setminus \{x_k\})$ the $(N+1) \times (N+1)$ Vandermonde matrix constructed with the nodal set $X_{N+1} \setminus \{x_k\}$. By using (2.3), the following is obtained for $k = 0, \ldots, N+1$:

$$\begin{aligned} c_k = \frac{\det V_k(X_{N+1})}{\det V(X_{N+1})} &= (-1)^{N+k+1} \varepsilon_{N+1} \frac{\det V(X_{N+1} \setminus \{x_k\})}{\det V(X_{N+1})} \\ &= (-1)^{N+k+1} \varepsilon_{N+1} \left(\prod_{\substack{0 \leq i < j \leq N+1 \\ i,j \neq k}} (x_j - x_i) \right) \Bigg/ \left(\prod_{0 \leq i < j \leq N+1} (x_j - x_i) \right) \\ &= \varepsilon_{N+1} \Bigg/ \left(\prod_{\substack{j=0 \\ j \neq k}}^{N+1} (x_k - x_j) \right). \end{aligned} \tag{3.4}$$

The denominator of this expression can be written as $\ell'_N(x_k)$, where $\ell_N(x) = \prod_{j=0}^{N}(x-x_j)$ is the nodal polynomial. To keep the dependence on x_{N+1} clear, this notation is used sparingly in this article.

The goal is to have positive weights, i.e. $w_k^{(N+1)} = w_k^{(N)} + c_k \geq 0$, which can be used to prove the following theorem.

Theorem 2. *Let X_N, W_N form an interpolatory quadrature rule. Then $X_{N+1} = X_N \cup \{x_{N+1}\}$ forms the nodal set of a positive interpolatory quadrature rule if and only if*

$$-\varepsilon_{N+1} \Bigg/ \left(\prod_{\substack{j=0 \\ j\neq k}}^{N+1} (x_k - x_j) \right) \leq w_k^{(N)}, \text{ for } k = 0, \ldots, N+1. \tag{3.5}$$

Proof. If X_{N+1} forms the nodal set of a positive interpolatory quadrature rule, then

$$0 \leq w_k^{(N+1)} = w_k^{(N)} + c_k = w_k^{(N)} + \varepsilon_{N+1} \Bigg/ \left(\prod_{\substack{j=0 \\ j\neq k}}^{N+1} (x_k - x_j) \right).$$

Subtracting $w_k^{(N)}$ from both sides of the inequality yields (3.5). Vice versa, if (3.5) holds, it follows that

$$w_k^{(N+1)} = w_k^{(N)} + c_k = w_k^{(N)} + \underbrace{\varepsilon_{N+1} \Bigg/ \left(\prod_{\substack{j=0 \\ j\neq k}}^{N+1} (x_k - x_j) \right)}_{=-w_k^{(N)}} = 0. \quad \square$$

If $\varepsilon_{N+1} = 0$, i.e. $\mathcal{A}_N x^{N+1} = \mu_{N+1}$, then the theorem yields that the new rule has positive weights if and only if the current rule has positive weights. This is not surprising: any node x_{N+1} can be added to such a rule with $w_{N+1}^{(N+1)} = 0$ (and with $w_k^{(N+1)} = w_k^{(N)}$ for $k = 0, \ldots, N$).

From a computational point of view (3.5) might not be a numerically stable way of computing the bounds that describe all nodes that can be added. In the context of quadrature rules, numerical instabilities are usually alleviated by changing the basis of the Vandermonde matrix, but this is not applicable in this case since the determinant is up to a scaling factor independent from the basis used to construct the Vandermonde matrix (and this factor cancels out in (3.3)). Nonetheless, (3.5) can be evaluated in a numerical stable way using the well-known barycentric formulation of the interpolating polynomial. The interested reader is referred to [28].

3.2. Quadrature rule adjustments

Theorem 2 describes a necessary and sufficient condition for a quadrature rule to have positive weights if both x_{N+1} and ε_{N+1} are known. A main novelty of this work is to employ a geometrical interpretation of (3.5), from which several possible adjustments of quadrature rules can be derived. The most straightforward one is that all nodes x_{N+1} can be determined that yield a positive interpolatory quadrature rule upon adding one of them to an existing quadrature rule. Moreover the formula also yields procedures to replace nodes in a quadrature rule, keeping the weights positive. The latter adjustment will be useful in Section 4, where it can be used to determine all possible M nodes that can be added to a rule.

In Section 3.2.1 we further consider (3.5) and discuss the geometrical relation between the new node x_{N+1} and the quadrature error ε_{N+1}. In Sections 3.2.2 and 3.2.3 we discuss the addition and replacement of nodes in a positive interpolatory quadrature rule such that positivity of the weights is preserved. These operations follow directly from the geometrical interpretation of Theorem 2 derived in Section 3.2.1. The removal of a node, as outlined in Section 2.3, can also be formulated as a consequence of Theorem 2, which is not done here since the removal of nodes has been considered extensively in previous work [19,22,23,26].

3.2.1. Geometry of nodal addition

The inequalities from (3.5) are $N+2$ linear inequalities in x_{N+1} and ε_{N+1}. This can be seen by rewriting (3.4) as follows:

$$c_k \prod_{\substack{j=0 \\ j\neq k}}^{N+1} (x_k - x_j) = \varepsilon_{N+1}, \text{ for } k = 0, \ldots, N+1. \tag{3.6}$$

If two values of x_{N+1}, c_k (for $k = 0, \ldots, N+1$), or ε_{N+1} are known, all other values can be determined from these expressions, which enforces that the obtained quadrature rule is again interpolatory. To incorporate positive weights, we

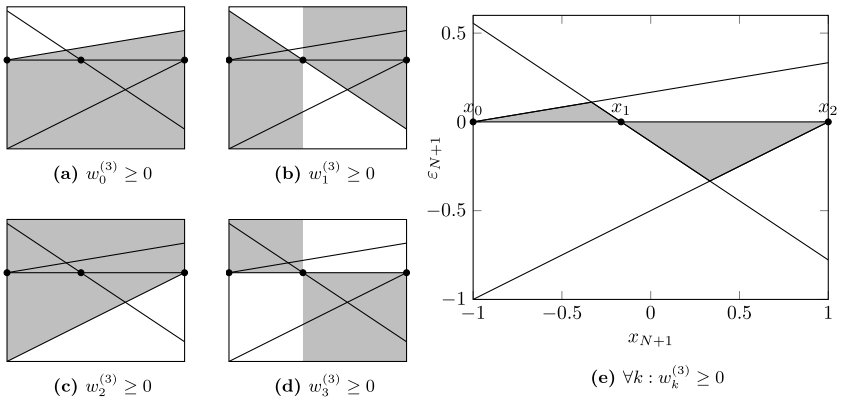


Fig. 1. The quadrature rule error ε_{N+1} versus the new node x_{N+1} using the quadrature rule $X_N = \{-1, -1/6, 1\}$ and $\rho \equiv 1/2$. The solid lines depict pairs $(x_{N+1}, \varepsilon_{N+1})$ such that one weight becomes zero, after addition of x_{N+1} to the quadrature rule using ε_{N+1} as quadrature error. *Left:* regions where individual weights are positive; the axes are labeled similar as the large rightmost figure. For example, if $(x_{N+1}, \varepsilon_{N+1})$ is picked in the gray region of Fig. 1(b), adding x_{N+1} to the quadrature rule yields a rule with $w_1^{(3)} \geq 0$ (assuming ε_{N+1} defines the raw moment correctly). *Right:* region where all weights are positive, which is the intersection of the left figures. Hence if $(x_{N+1}, \varepsilon_{N+1})$ is picked in the gray region of Fig. 1(e), adding x_{N+1} to the quadrature rule yields a rule with positive weights.

use that for $k = 0, \ldots, N$ it holds that

$$\varepsilon_{N+1} = c_k \prod_{\substack{j=0 \\ j\neq k}}^{N+1} (x_k - x_j) = (x_k - x_{N+1})\, c_k \underbrace{\prod_{\substack{j=0 \\ j\neq k}}^{N} (x_k - x_j)}_{\text{Independent from } x_{N+1}}.$$

By combining this with (3.5) and requiring $w_k^{(N)} + c_k \geq 0$ inequalities of the following form are obtained:

$$\begin{aligned} \varepsilon_{N+1} &\leq -w_k^{(N)}(x_k - x_{N+1}) \prod_{\substack{j=0 \\ j\neq k}}^{N} (x_k - x_j) \quad \text{if } \prod_{\substack{j=0 \\ j\neq k}}^{N+1} (x_k - x_j) \leq 0, \\ \varepsilon_{N+1} &\geq -w_k^{(N)}(x_k - x_{N+1}) \prod_{\substack{j=0 \\ j\neq k}}^{N} (x_k - x_j) \quad \text{if } \prod_{\substack{j=0 \\ j\neq k}}^{N+1} (x_k - x_j) \geq 0. \end{aligned} \tag{3.7}$$

These are linear inequalities describing the relation between x_{N+1} and ε_{N+1} such that $w_k^{(N+1)} \geq 0$ for $k = 0, \ldots, N$. For $k = N + 1$ it holds that $w_k^{(N)} = 0$, so by using that $c_k = w_k^{(N+1)}$, (3.6) translates to:

$$\begin{aligned} \varepsilon_{N+1} &\leq 0 \quad \text{if } \prod_{j=0}^{N} (x_{N+1} - x_j) \leq 0, \\ \varepsilon_{N+1} &\geq 0 \quad \text{if } \prod_{j=0}^{N} (x_{N+1} - x_j) \geq 0. \end{aligned} \tag{3.8}$$

Even though the rightmost inequalities are non-linear, their sign solely depends on the location of x_{N+1} with respect to the other nodes. Hence the exact value of the product is not of importance.

Example 1. The inequalities from (3.8) are visualized as functions from x_{N+1} to ε_{N+1} in Fig. 1 for the quadrature rule with X_N and W_N as follows:

$$X_N = \left\{-1, -\frac{1}{6}, 1\right\}, W_N = \left\{\frac{1}{10}, \frac{24}{35}, \frac{3}{14}\right\}.$$

This is an (obviously positive) interpolatory quadrature rule with $\Omega = [-1, 1]$ and $\rho \equiv 1/2$. The solid lines in the figures depict all $(x_{N+1}, \varepsilon_{N+1})$ pairs such that one weight becomes equal to zero (i.e. where equality is attained in inequality (3.7) or (3.8)). The region where individual weights are positive is shaded in Figs. 1(a), 1(b), 1(c), and 1(d). Fig. 1(e) is the intersection of these figures and therefore depicts regions where all weights are positive. Any $(x_{N+1}, \varepsilon_{N+1})$ pair in the shaded region describes a positive interpolatory quadrature rule that contains the original three nodes.

The left subfigures demonstrate some key properties of the derived inequalities. The inequalities are linear and switch sign at the node, which is the rightmost condition of (3.7). The characteristics of the last inequality (Fig. 1(d)) solely depend on the location of x_{N+1} with respect to the other nodes. A combination of all inequalities (Fig. 1(e)) has varying characteristics between different nodes, but it is always a system of linear inequalities. The line $\varepsilon_{N+1} = 0$ is contained in all shaded regions, because any node with weight equal to zero can be added to the rule if the next moment μ_{N+1} is already correctly integrated by the quadrature rule.

The relation between x_{N+1} and ε_{N+1} from (3.8) can be interpreted in two ways. Firstly, if a new node x_{N+1} is given, an upper bound and a lower bound on ε_{N+1} can be determined such that upon adding x_{N+1} to the quadrature rule, a positive interpolatory quadrature rule is obtained. Geometrically these are the bounds of the shaded area with the $x = x_{N+1}$ line. This interval is never empty (as $\varepsilon_{N+1} = 0$ is always in the shaded region). Secondly, if ε_{N+1} is given, a (possibly empty) set can be determined such that a positive interpolatory quadrature rule is obtained upon adding a node from such a set. Geometrically this is equivalent to determining the bounds of the shaded area with the $y = \varepsilon_{N+1}$ line.

The second interpretation can be used to add nodes to a quadrature rule, i.e. ε_{N+1} is known and the goal is to determine x_{N+1} (this is discussed in Section 3.2.2). The first interpretation can be used to replace nodes within a quadrature rule: x_{N+1} is added to the nodal set and an existing node can be removed by setting its weight to zero (this is discussed in Section 3.2.3).

3.2.2. Addition of a node

A direct consequence of (3.7) is that all nodes that can be added to a quadrature rule can be defined by means of intervals, obtained via a linear inequality. The results are discussed in the following lemmas. The first focuses on keeping the *existing* weights of the quadrature rule positive, the second focuses on ensuring that the *additional* weight (i.e. of the added node) is positive.

Lemma 3. *Let X_N, W_N form the nodes and the weights of a positive interpolatory quadrature rule, let ε_{N+1} from (3.2) be given, and let index k of node x_k be given. Let $x_{N+1}^{[k]}$ be as follows:*

$$x_{N+1}^{[k]} = \left(\varepsilon_{N+1} + w_k^{(N)} x_k \ell_N'(x_k)\right) \Big/ \left(w_k^{(N)} \ell_N'(x_k)\right).$$

Then $w_k^{(N+1)} \geq 0$ if and only if $x_{N+1} \in I_k$ with

$$I_k = \mathbb{R} \setminus [x_k, x_{N+1}^{[k]}) \text{ if } x_k < x_{N+1}^{[k]}, \text{ and } I_k = \mathbb{R} \setminus (x_{N+1}^{[k]}, x_k] \text{ otherwise.}$$

Or in other words, if and only if x_{N+1} is not between x_k and $x_{N+1}^{[k]}$.

Proof. Adding a node is determining an x_{N+1} that solves (3.7) if ε_{N+1} is known. Hence, to keep the kth weight positive, this is equivalent to computing the solution $x_{N+1}^{[k]}$ of the following problem:

$$\varepsilon_{N+1} = -w_k^{(N)}(x_k - x_{N+1}^{[k]}) \underbrace{\prod_{\substack{j=0 \\ j\neq k}}^{N} (x_k - x_j)}_{\ell_N'(x_k)}, \tag{3.9}$$

Here we used ℓ_N' to make the notation more compact. Hence if $w_k^{(N)} \neq 0$:

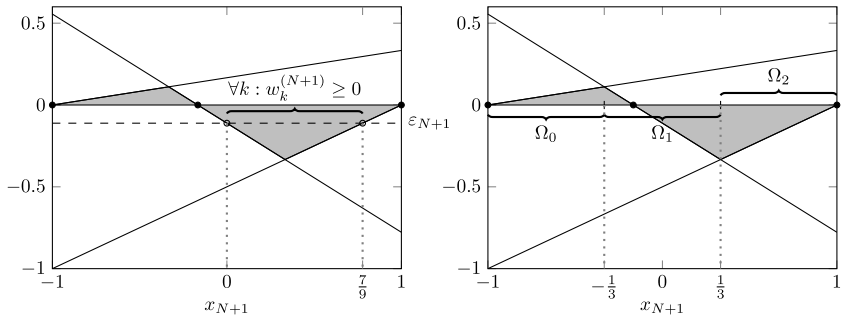
$$x_{N+1}^{[k]} = \frac{\varepsilon_{N+1} + w_k^{(N)} x_k \ell_N'(x_k)}{w_k^{(N)} \ell_N'(x_k)}.$$

The node $x_{N+1}^{[k]}$ is such that, if added to the quadrature rule, an interpolatory quadrature rule is obtained with $w_k^{(N+1)} = 0$ (the other weights may be negative). Assume $x_k < x_{N+1}^{[k]}$, without loss of generality. Then any node x_{N+1} with $x_{N+1} \geq x_{N+1}^{[k]}$ or $x_{N+1} < x_k$ solves (3.7) for a single k. This is equivalent to stating that $x_{N+1} \in I_k = \mathbb{R} \setminus [x_k, x_{N+1}^{[k]})$. □

The proof of this lemma can also be stated geometrically, using one of Figs. 1(a), 1(b), or 1(c). If ε_{N+1} is known, those x_{N+1} that are such that $(x_{N+1}, \varepsilon_{N+1})$ is not part of a gray region form the interval as stated in the lemma. Here, $x_{N+1}^{[k]}$ is the intersection of the line passing through x_k and the constant line ε_{N+1}. All intervals I_k are bounded, so there always exists a node $x_{N+1} \in (I_0 \cap \cdots \cap I_N)$, or in other words, there always exists a node that keeps the existing $N+1$ weights of a quadrature rule positive upon addition.

Obviously, the goal is also to ensure that the weight of the added node is positive, which can be described by means of a series of intervals. The details of this are discussed in the following lemma.

Lemma 4. *Let X_N, W_N form the nodes and the weights of a positive interpolatory quadrature rule, let ε_{N+1} from (3.2) be given. Without loss of generality, assume that $x_0 < x_1 < \cdots < x_N$. Then $w_{N+1}^{(N+1)} \geq 0$ upon addition of x_{N+1} to the quadrature rule if and only if one of the following holds for all $k = 0, \ldots, N$:*



(a) Addition ($\varepsilon_{N+1} = -1/9$) (b) Replacement

Fig. 2. Addition of a new node to and replacement of an existing node within the quadrature rule $X_N = \{-1, -1/6, 1\}$ and $\rho \equiv 1/2$. *Left:* all nodes that can be added to a quadrature rule form intervals, in this case the interval $[0, 7/9]$ and the interval $(-\infty, -5/3]$ (of which the latter is not depicted). *Right:* the closed sets Ω_k depict all possible replacements within a quadrature rule. If the goal is to construct a positive interpolatory quadrature rule, the node x_k can only be replaced by nodes from the set Ω_k.

- $x_{N+1} \in [x_{k-1}, x_k]$ *if the signs of* $\ell_N'(x_k)$ *and* ε_{N+1} *are equal (e.g. both are negative);*
- $x_{N+1} \in [x_k, x_{k+1}]$ *if the signs of* $\ell_N'(x_k)$ *and* ε_{N+1} *differ.*

For $k = N$*, use* $x_{k+1} = \infty$ *and for* $k = 0$*, use* $x_{k-1} = -\infty$ *(with a little abuse of notation).*

Proof. Recall the derivation of (3.8), i.e. the relation between x_{N+1}, $w_{N+1}^{(N+1)}$, and ε_{N+1}:

$$-w_{N+1}^{(N+1)} \prod_{j=0}^{N} (x_{N+1} - x_j) = \varepsilon_{N+1}.$$

It holds that $w_{N+1}^{(N+1)} > 0$ if $\prod_{j=0}^{N}(x_{N+1} - x_j)$ and ε_{N+1} have different sign. The first term flips sign only at $x_{N+1} = x_k$ (for any $k = 0, \ldots, N$), hence if, for given k,

$$\prod_{\substack{j=0 \\ j \neq k}}^{N} (x_k - x_j) > 0,$$

it is necessary that $x_{k-1} < x_{N+1} < x_k$ to ensure that $\prod_{j=0}^{N}(x_{N+1} - x_j)$ is negative and $x_k < x_{N+1} < x_{k+1}$ to ensure that $\prod_{j=0}^{N}(x_{N+1} - x_j)$ is positive. A similar result holds if

$$\prod_{\substack{j=0 \\ j \neq k}}^{N} (x_k - x_j) < 0.$$

Combining this with the sign of ε_{N+1} results in the statement of the lemma. □

Geometrically, Lemma 4 describes the intervals of Fig. 1(d). Notice that Lemma 4 can also straightforwardly be applied to cases where $w_k^{(N)} = 0$ (for any $k = 0, \ldots, N$), i.e. if the quadrature rule has weights equal to zero.

Using Lemmas 3 and 4 the set I can be computed such that any $x_{N+1} \in I$ can be added to a quadrature rule X_N and W_N such that positive weights are obtained (and adding any $x_{N+1} \notin I$ yields a rule with at least one negative weight). The procedure is to firstly compute all intervals $I_0, \ldots, I_N$ from Lemma 3 and construct $I = I_0 \cup \cdots \cup I_N$. Secondly, Lemma 4 is used to remove intervals of the form $[x_{k-1}, x_k]$ from I.

The exact details of this procedure are outlined in Algorithm 1. No advanced interval arithmetic is necessary to implement this algorithm, only a procedure that implements the removal of an interval from a series of intervals is needed.

Example 2. Reconsider the quadrature rule from Example 1. Then the bounds of the intervals containing nodes that can be added, i.e. the solutions of (3.9), are depicted in Fig. 2(a) as open circles. Here, $\mu_{N+1} = 0$, so from a straightforward computation it follows that $\varepsilon_{N+1} = -1/9$. A constant ρ is considered here. In this case, the values of $x_{N+1}^{[k]}$ are (from left to right) $-5/3$, 0, and $7/9$, of which the first is not visible in the figure. Adding any of these nodes yields a quadrature rule with positive weights, but we emphasize that this is generally not the case for other quadrature rules. Hence adding any node from the set $I = (-\infty, -5/3] \cup [0, 7/9]$ yields a positive interpolatory quadrature rule. Restricting x_{N+1} to the set Ω further reduces the number of possible intervals.

Algorithm 1 Addition of a node

Input: Positive, interpolatory quadrature rule X_N, W_N, raw moment μ_{N+1} (or, equivalently, ε_{N+1})
Output: Set $I \subset \mathbb{R}$ such that $X_N \cup \{x\}$ forms the nodal set of a positive, interpolatory quadrature rule if and only if $x \in I$

$I \leftarrow \mathbb{R}$
$\varepsilon_{N+1} \leftarrow \mu_{N+1} - \sum_{k=0}^{N} x_k^{N+1} w_k^{(N)}$
Sort X_N, W_N such that $x_0 < x_1 < \cdots < x_N$
for $k = 0, \ldots, N+1$ **do**
    $\ell_N'(x_k) \leftarrow \prod_{j \neq k}^{N} (x_k - x_j)$
    if $w_k^{(N)} > 0$ **then**
        $x_{N+1}^{[k]} \leftarrow \left(\varepsilon_{N+1} + w_k^{(N)} x_k \ell_N'(x_k)\right) \Big/ \left(w_k^{(N)} \ell_N'(x_k)\right)$
        if $x_{N+1}^{[k]} > x_k$ **then**
            $I \leftarrow I \setminus [x_k, x_{N+1}^{[k]})$
        else
            $I \leftarrow I \setminus (x_{N+1}^{[k]}, x_k]$
        end if
    end if
    if ($\ell_N'(x_k) < 0$ and $\varepsilon_{N+1} < 0$) or ($\ell_N'(x_k) > 0$ and $\varepsilon_{N+1} > 0$) **then**
        if $k > 0$ and $w_k > 0$ **then**
            $I \leftarrow I \setminus [x_{k-1}, x_k]$
        else
            $I \leftarrow I \setminus (-\infty, x_k]$
        end if
    else
        if $k < N$ and $w_k > 0$ **then**
            $I \leftarrow I \setminus [x_k, x_{k+1}]$
        else
            $I \leftarrow I \setminus [x_k, \infty)$
        end if
    end if
end for
Return I

Notice that $I = \emptyset$ if $\varepsilon_{N+1} \neq 0$ and $w_k^{(N)} = 0$. This can be derived mathematically, but it also follows from the mere fact that all weights change (see (3.9)) upon addition of a node to a quadrature rule, so $w_k^{(N+1)} = w_k^{(N)} = 0$ is not possible. If $\varepsilon_{N+1} = 0$, no node can be added to enforce that $w_k^{(N)} = 0$. However, any node with weight equal to zero can be added, hence the formula yields $x_{N+1}^{[k]} = x_k$ with $w_{N+1}^{(N+1)} = 0$. Technically, the quadrature rule now has a node equal to x_k with weight equal to zero. Nonetheless, this results in a singular Vandermonde matrix (which contradicts the theory developed so far), so we do not further study this specific case.

If $\Omega = \mathbb{R}$ and the number of nodes is odd, it is always possible to add a single node to a quadrature rule: in this case the result from Lemma 4 either states that $x_{N+1} \in (-\infty, x_0]$ or $x_{N+1} \in [x_N, \infty)$, but never both. Geometrically this means that the leftmost and rightmost shaded regions grow to infinity and minus infinity respectively (or vice versa). Similarly, if $\Omega = \mathbb{R}$ and the number of nodes is even, it is always possible to add a single node if $\varepsilon_{N+1} \geq 0$.

However, in any other case (i.e. that of a bounded Ω or even number of nodes with $\varepsilon_{N+1} < 0$) adding a single node to a quadrature rule is not always possible, as shown in the following example.

Example 3. Adding a single node to the following interpolatory quadrature rule is not possible when requiring positive weights:

$$X_N = \left\{-1, -\frac{1}{6}, \frac{1}{11}, 1\right\}, W_N = \left\{\frac{29}{180}, \frac{144}{595}, \frac{1331}{3060}, \frac{17}{105}\right\}.$$

Note that this example can be obtained straightforwardly by adding the node $1/11$ to the quadrature rule from Example 1 and redetermining the weights likewise.

3.2.3. Replacement of a node

Replacing a node is equivalent to adding a node, with the difference that the goal is to determine this node such that the weight of an existing node in the obtained quadrature rule becomes zero, i.e. $w_k^{(N+1)} = 0$ for a $k \leq N$. This is equivalent to determining a specific $(x_{N+1}, \varepsilon_{N+1})$ pair that yields $w_k^{(N+1)} = 0$, which was used to determine all possible additions in Section 3.2.2. The main difference with addition is that the next moment μ_{N+1} is not used, as the number of nodes and the degree of the rule do not change. This makes ε_{N+1} a free variable.

The relation between ε_{N+1} and x_{N+1} is already derived, so by reconsidering (3.9) with the goal to determine both x_{N+1} and all $\varepsilon_{N+1}^{[k]}$ (indexed by $[k]$ with $k = 0, \ldots, N$) that make $w_k^{(N+1)} = 0$ the following expressions are obtained:

$$\varepsilon_{N+1}^{[k]} = -w_k^{(N)}(x_k - x_{N+1}) \prod_{\substack{j=0 \\ j\neq k}}^{N} (x_k - x_j), \quad \text{for } k = 0, \ldots, N. \tag{3.10}$$

We will interpret this expression as a function of x_{N+1}, denoted by $\varepsilon_{N+1}^{[k]} : \Omega \to \mathbb{R}$. By using $\varepsilon_{N+1} = \varepsilon_{N+1}^{[k]}(x_{N+1})$, a positive interpolatory quadrature rule with $w_k^{(N+1)} = 0$ is obtained upon adding x_{N+1} to the rule.

It follows that for every $x_{N+1} \in \Omega$ there is an $x_k \in X_N$ such that the quadrature rule with nodes $(X_N \cup \{x_{N+1}\}) \setminus \{x_k\}$ is positive and interpolatory. The details are discussed in the following lemma.

Lemma 5. *Let X_N, W_N form the nodes and the weights of a positive interpolatory quadrature rule and let $x_{N+1} \in \Omega$ be given. Then there exists an x_k such that $(X_N \cup \{x_{N+1}\}) \setminus \{x_k\}$ forms the nodal set of a positive and interpolatory quadrature rule.*

Proof. Let $\varepsilon_{N+1}^{[k]}$ be defined by (3.10). Consider ε_- and ε_+, defined as follows:

$$\varepsilon_- = \max_k \left(\varepsilon_{N+1}^{[k]} \mid \varepsilon_{N+1}^{[k]} < 0 \right),$$
$$\varepsilon_+ = \min_k \left(\varepsilon_{N+1}^{[k]} \mid \varepsilon_{N+1}^{[k]} > 0 \right).$$

Hence $\varepsilon_- < 0 < \varepsilon_+$. Using Lemma 3, it follows that using either ε_- or ε_+ to add x_{N+1} results in a quadrature rule with $w_k^{(N+1)} \geq 0$ for $k = 0, \ldots, N$. Moreover, by definition of ε_- and ε_+ these rules have one (or more) weight equal to 0. From Lemma 4 it follows that either the rule constructed using ε_- or the rule constructed using ε_+ has $w_{N+1}^{(N+1)} \geq 0$ (and the other has $w_{N+1}^{(N+1)} \leq 0$).

Concluding, either ε_- or ε_+ can be used to construct a positive interpolatory quadrature rule with at least one weight equal to zero. Nodes with weights equal to zero can be removed without affecting the quadrature rules. This is equivalent to having added a node x_{N+1} and having removed one, say x_k, which is the statement of the lemma. □

The proof of the lemma is constructive, and therefore describes a straightforward method to replace nodes in a quadrature rule. Given $x_{N+1} \in \Omega$, the procedure is to compute ε_+ and ε_-, figure out whether $\varepsilon_{N+1} = \varepsilon_+$ or $\varepsilon_{N+1} = \varepsilon_-$ yields $w_{N+1}^{(N+1)} \geq 0$ by using Lemma 4, and finally compute the quadrature rule after replacement. These steps are outlined in detail in Algorithm 2. Geometrically, the approach computes the two lines closest to the $\varepsilon_{N+1} = 0$ line, i.e. the boundary of the gray region, and determines which of these lines correspond to obtaining a quadrature rule with only positive weights (see Fig. 2(b)).

Consequently, the domain of a quadrature rule, depicted by $\Omega \subset \mathbb{R}$, can be decomposed in subsets $\Omega_0, \ldots, \Omega_N$ that indicate which node can be replaced. If $x_{N+1} \in \Omega_k$, $(X_N \cup \{x_{N+1}\}) \setminus \{x_k\}$ forms the nodal set of a positive interpolatory quadrature rule. Combining the results of Lemmas 4 and 5, these sets can be denoted in the following way:

$$x_{N+1} \in \Omega_k \iff \begin{cases} \varepsilon_{N+1}^{[k]} = \min_j \left(\varepsilon_{N+1}^{[j]} \mid \varepsilon_{N+1}^{[j]}(x_{N+1}) \geq 0 \right) & \text{if } \prod_{j\neq k}^{N+1} (x_k - x_j) \leq 0, \\ \varepsilon_{N+1}^{[k]} = \max_j \left(\varepsilon_{N+1}^{[j]} \mid \varepsilon_{N+1}^{[j]}(x_{N+1}) \leq 0 \right) & \text{if } \prod_{j\neq k}^{N+1} (x_k - x_j) \geq 0. \end{cases}$$

The sets Ω_k have been depicted in Fig. 2(b). Notice that the boundaries of these sets correspond to positions where two lines intersect, or in other words, those $x_{N+1} \in \Omega_k$ that result in two weights equal to zero, if used for replacement. One of these weights is, by construction, $w_k^{(N+1)}$. If the other weight is $w_l^{(N+1)}$, we also have $x_{N+1} \in \Omega_l$. This geometrical observation can be made explicit, which can be used to actually compute Ω_k: these x_{N+1} have $\varepsilon_{N+1}^{[k]}(x_{N+1}) = \varepsilon_{N+1}^{[l]}(x_{N+1})$, or equivalently:

$$-w_k^{(N)}(x_k - x_{N+1}) \prod_{\substack{j=0 \\ j\neq k}}^{N} (x_k - x_j) = -w_l^{(N)}(x_l - x_{N+1}) \prod_{\substack{j=0 \\ j\neq l}}^{N} (x_l - x_j).$$

Hence we have proved the following lemma.

Algorithm 2 Replacement of a new node

Input: Positive, interpolatory quadrature rule X_N, W_N, new node $x \notin X_N$
Output: Positive, interpolatory quadrature rule $\hat{X}_N$, $\hat{W}_N$, with $x \in \hat{X}_N$ and $\#(\hat{X}_N \cap X_N) = N$

$\varepsilon_+ \leftarrow \infty$
$\varepsilon_- \leftarrow -\infty$
$k_+, k_- \leftarrow -1$
for $k = 0, \ldots, N$ **do**
  $\varepsilon_{N+1}^{[k]} \leftarrow -w_k^{(N)}(x_k - x_{N+1}) \prod_{j\neq k}^{N}(x_k - x_j)$
  $\ell'_{N+1} = (x - x_j) \prod_{j\neq k}^{N}(x_k - x_j)$
  if $\ell'_{N+1} \leq 0$ and $\varepsilon_+ > \varepsilon_{N+1}^{[k]} > 0$ **then**
    $k_+ \leftarrow k$
    $\varepsilon_+ \leftarrow \varepsilon_{N+1}^{[k]}$
  end if
  if $\ell'_{N+1} \geq 0$ and $\varepsilon_- < \varepsilon_{N+1}^{[k]} < 0$ **then**
    $k_- \leftarrow k$
    $\varepsilon_- \leftarrow \varepsilon_{N+1}^{[k]}$
  end if
end for

if $\prod_{j=0}^{N}(x - x_j) > 0$ **then**
  $c_k \leftarrow \varepsilon_+ \Big/ \Big((x - x_j) \prod_{j\neq k}^{N}(x_k - x_j)\Big)$ (for $k = 0, \ldots, N$)
  $c \leftarrow \varepsilon_+ \Big/ \prod_{j=0}^{N}(x - x_j)$
  $k_0 \leftarrow k_+$
else
  $c_k \leftarrow \varepsilon_- \Big/ \Big((x - x_j) \prod_{j\neq k}^{N}(x_k - x_j)\Big)$ (for $k = 0, \ldots, N$)
  $c \leftarrow \varepsilon_- \Big/ \prod_{j=0}^{N}(x - x_j)$
  $k_0 \leftarrow k_-$
end if

$\hat{X}_N \leftarrow \{x_0, \ldots, x_{k_0-1}, x, x_{k_0+1}, \ldots, x_N\}$
$\hat{W}_N \leftarrow \{w_0^{(N)} + c_0, \ldots, w_{k_0-1}^{(N)} + c_{k_0-1}, c, w_{k_0+1}^{(N)} + c_{k_0+1}, \ldots, w_N^{(N)} + c_N\}$
Return $\hat{X}_N$, $\hat{W}_N$

Lemma 6. *Let k be given and let $\partial\Omega_k$ denote the boundary of Ω_k. Then, for any $x_{N+1} \in \partial\Omega_k$, we have that*

$$-w_k^{(N)}(x_k - x_{N+1}) \prod_{\substack{j=0 \\ j\neq k}}^{N}(x_k - x_j) = -w_l^{(N)}(x_l - x_{N+1}) \prod_{\substack{j=0 \\ j\neq l}}^{N}(x_l - x_j),$$

for an $l \in 0, \ldots, N$.

The result is a procedure to compute the boundaries of a specific Ω_k. Firstly, for $l = 0, \ldots, N$, compute $x_{(k,l)}$ such that

$$-w_k^{(N)}(x_k - x_{(k,l)}) \prod_{\substack{j=0 \\ j\neq k}}^{N}(x_k - x_j) = -w_l^{(N)}(x_l - x_{(k,l)}) \prod_{\substack{j=0 \\ j\neq l}}^{N}(x_l - x_j). \tag{3.11}$$

Those $x_{(k,l)}$ that yield a positive interpolatory quadrature rule upon replacement (e.g. computed using Algorithm 2), form the boundary of the interval Ω_k. If $x_l < x_{(k,l)}$, it follows that $[x_l, x_{(k,l)}) \notin \Omega_k$, since a replacement with $x_{N+1} \in [x_l, x_{(k,l)}]$ results in a negative $w_l^{(N+1)}$ (similar for $x_l > x_{(k,l)}$). The procedure to determine Ω_k explicitly is outlined in Algorithm 3. Here, the indexing is slightly changed to be able to reuse parts of Algorithm 1, since we still need to ensure that the weight of the added node (which replaces x_k) is positive.

Algorithm 3 Replacement of a given node

Input: Positive, interpolatory quadrature rule X_N, W_N, node $x_l \in X_N$
Output: Space Ω_l, such that $(X_N \cup \{x\}) \setminus \{x_l\}$ forms the nodal set of a positive, interpolatory quadrature rule if and only if $x \in \Omega_l$

$\Omega_l \leftarrow \mathbb{R}$
$\ell'_N(x_l) \leftarrow \prod_{j\neq l}^{N}(x_l - x_j)$
for $k = 0, \dots, l-1, l+1, \dots, N$ **do**
  $\ell'_N(x_k) \leftarrow \prod_{j\neq k}^{N}(x_k - x_j)$
  $x_{(k,l)} \leftarrow \left(w_k^{(N)} x_k \ell'_N(x_k) - w_l^{(N)} x_l \ell'_N(x_l)\right) \Big/ \left(w_k \ell'_N(x_k) - w_l \ell'_N(x_l)\right)$
  if $x_k < x_{(k,l)}$ **then**
    $\Omega_l \leftarrow \Omega_l \setminus [x_k, x_{(k,l)})$
  else
    $\Omega_l \leftarrow \Omega_l \setminus (x_{(k,l)}, x_k]$
  end if
  $\varepsilon_{N+1} \leftarrow -w_l(x_l - x_{(k,l)})\ell'_N(x_l)$
  Follow steps 14–26 of Algorithm 1 with $I = \Omega_l$
end for

Eq. (3.11) does not necessarily have a solution for any l. Geometrically this is the case if the lines through x_k and x_l are parallel. In such a case, one should use $x_{(k,l)} \leftarrow \infty$ or $x_{(k,l)} \leftarrow -\infty$ in Algorithm 3, depending on the sign of the nominator when computing $x_{(k,l)}$ (usually, this happens automatically when using floating point arithmetic).

The values of x_{N+1} that solve (3.11) form a special case. Since $x_{N+1} \in \Omega_k \cap \Omega_l$, the quadrature rule $(X_N \cup \{x_{N+1}\}) \setminus \{x_k, x_l\}$ is positive, interpolatory, and has degree N, even though it consists only of N nodes. The latter result is remarkable: two nodes are removed and one is added, but the degree of the quadrature rule is not affected. Such rules have a non-trivial high degree and are therefore more accurate than interpolatory quadrature rules without this property.

Example 4. An example of an interpolatory quadrature rule with non-trivial high degree is $X_N = \{-1, 1/3\}$, obtained by adding $1/3$ to the quadrature rule of Example 1 (and removing all nodes with zero weight). All nodes that can be added to obtain such a rule are the intersection of two lines in Fig. 2(b).

More generally, all nodes x_k that can be added to a rule can be found by determining the bounds of the shaded region and observing which node belongs to the obtained bound. Consequently, the fact that $\Omega = \bigcup_{k=0}^{N} \Omega_k$ follows visually from Fig. 2(b). Hence the relation between $\varepsilon_{N+1}^{[k]}$ and x_{N+1}, as described by (3.10), are the solid lines in Fig. 2(b).

The node $x_{(k,l)}$ only depends on the nodes x_j with $j \neq k$ and $j \neq l$, i.e. its value is independent from x_k and x_l. This is not evident, as (3.11) depends on these nodes. However, it can be demonstrated by using that the rule is interpolatory, which yields:

$$w_k^{(N)} = \int_\Omega L_k(x)\,\rho(x)\,\mathrm{d}x = \frac{1}{\ell'_N(x_k)} \int_\Omega \frac{\ell_N(x)}{x - x_k}\rho(x)\,\mathrm{d}x, \quad \text{with } L_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{N} \frac{x - x_j}{x_k - x_j}.$$

Here, $L_k(x)$ is the kth Lagrange basis polynomial. Replacing this expression in (3.11) and using that $\ell'_N(x_k) = \prod_{j\neq k}(x_k - x_j)$ yield an equality that can be simplified to the following:

$$x_{(k,l)} = \left(\int_\Omega x\,\ell_{(k,l)}(x)\,\rho(x)\,\mathrm{d}x\right) \Big/ \left(\int_\Omega \ell_{(k,l)}(x)\,\rho(x)\,\mathrm{d}x\right), \quad \text{with } \ell_{(k,l)}(x) = \prod_{\substack{j=0 \\ j\neq k,l}}^{N} (x - x_j).$$

This expression is in fact a Patterson extension (consider (2.6) with $j = 0$). The tight relation between the Patterson extension and the framework discussed in this article is further discussed in Section 3.3.1.

3.3. Constructing quadrature rules

In the previous section the theoretical foundation for extending a positive interpolatory quadrature rule with a single node is derived. In this section, firstly it is discussed how addition relates naturally to the Patterson extension [8,29] of (non-Gaussian) quadrature rules. Secondly, due to the simplicity of addition and replacement of a node, quadrature rules based on these procedures can be derived numerically fast and accurately, and an example is discussed.

As discussed previously, there does not always exist a single node that can be added such that positive weights are obtained, so it is non-trivial to construct a sequence of positive interpolatory quadrature rules by consecutively adding a single node to the rule. There are various possibilities to alleviate this, e.g. by allowing negative weights, relaxing the strict requirement that all nodes of the quadrature rule have to be preserved, or by adding multiple nodes instead of one. In this article, the second and third options are further considered. For this purpose, a quadrature rule is presented based on the *replacement* of nodes. The rule has positive weights and is interpolatory, but is strictly speaking not fully nested. The details are considered in Section 3.3.2. The addition of multiple nodes is further discussed in Section 4.

3.3.1. Patterson extension

Remarkably, both the addition and replacement of a node can yield a Patterson extension of a quadrature rule. In both cases, the focus is on the nodes that yield a zero weight upon addition to the quadrature rule.

In Section 3.2.2 it was noticed that any weight from a quadrature rule can be made equal to zero by exploiting the relation between ε_{N+1} and x_{N+1}. In Example 2 the quadrature rule $X_N = \{-1, -1/6, 1\}$ was considered, where the nodes $-5/3$, 0, and $7/9$ are such that upon adding one of these to the rule, a rule of only three nodes with non-zero weights of degree three is obtained. Notice that these nodes are Patterson extensions of quadrature rules (as discussed in Section 2.2), as they can be interpreted as adding one node ($M = 1$) to a quadrature rule of two nodes ($N = 1$), obtaining a rule of degree three ($N + 2M = 3$). This also holds in general: for given k, adding one node $x_{N+1}^{[k]}$ from (3.9) (so $M = 1$) to the interpolatory quadrature rule $X_N \setminus \{x_k\}$ (with degree $N - 1$) yields a quadrature rule with $N + 1$ nodes and degree $N + 1$ (which equals $(N - 1) + 2M$).

In Section 3.2.3 the notation $x_{(k,l)}$ was introduced to denote nodes that, upon adding them to the rule, yield a (possibly negative) interpolatory quadrature rule with $w_k^{(N+1)} = w_l^{(N+1)} = 0$. These nodes also form a Patterson extension. To see this, notice that the replacement is adding a single node to the quadrature rule $X_{N-2} = X_N \setminus \{x_k, x_l\}$. The Patterson extension of a single node of this quadrature rule is a quadrature rule consisting of N nodes of degree $(N - 2) + 2M = N$ (adding one node means $M = 1$). By construction, this rule has the nodes $X_{N-2} \cup \{x_{(k,l)}\}$.

Example 5. Reconsider for example the quadrature rule with the nodes $X_N = \{-1, -1/6, 1\}$ and $\rho \equiv 1/2$. Then it is straightforward to determine using (3.11) that $x_{(0,1)} = -1/3$, $x_{(0,2)} = 2$, and $x_{(1,2)} = 1/3$. Hence these are three Patterson extensions of the quadrature rule nodes $\{1\}$, $\{-1/6\}$, and $\{-1\}$. Indeed, the quadrature rules with the nodes $\{-1/3, 1\}$, $\{-1/6, 2\}$, or $\{-1, 1/3\}$ have degree equal to 2.

Notice that $x_{(k,l)}$ is not a Patterson extension of the quadrature rule that has been used to determine it, i.e. X_N, W_N in (3.11). However, its definition allows for a straightforward way to determine this extension. First, add (randomly) two nodes to the quadrature rule X_N, W_N, obtaining a possibly negative interpolatory quadrature rule X_{N+2}, W_{N+2}. Then the node $x_{(N+1,N+2)}$ is the Patterson extension of the quadrature rule with nodes X_N, because upon adding this node to X_{N+2}, the weights of the randomly added nodes become zero. As the Patterson extension is unique, this construction is well-defined. Naturally, this is not the preferred approach to construct a Patterson extension, but it embeds such extensions into the framework discussed here.

The Patterson extension is also obtained as a special case if multiple nodes are added to a quadrature rule. This will be discussed in Section 4.3.

3.3.2. Partially nested, positive, and interpolatory quadrature rule

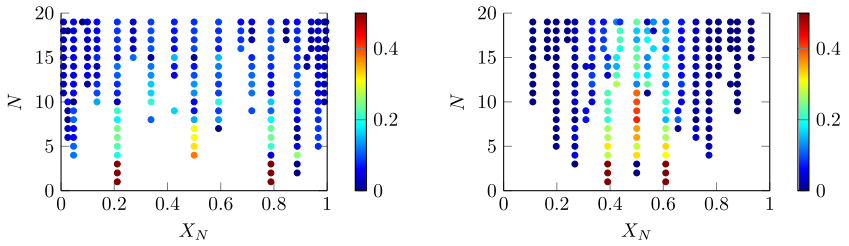
The addition and replacement of a single node are straightforward procedures described as the solutions of linear inequalities. However, there does not always exist a single node that can be added such that all weights remain positive. In this section, this is alleviated by relaxing the requirement that $X_N \subset X_{N+M}$.

To this end, let X_N and $\hat{X}_{N+1}$ be the nodes of two positive interpolatory quadrature rules, possibly with $X_N \not\subset \hat{X}_{N+1}$. The nodes $\hat{X}_{N+1}$ can for example form a Gaussian quadrature rule. The idea is to iteratively replace nodes in $\hat{X}_{N+1}$ with nodes from X_N, i.e. removing $x_k \in \hat{X}_{N+1}$ and adding $x_k \in X_N$. Ideally, all nodes $x_k \in X_N$ can be added to $x_k \in \hat{X}_{N+1}$, which would yield a rule that reuses all nodes in X_N.

In other words, if $X_N = \{x_0, \ldots, x_N\}$ and $\hat{X}_{N+1} = \{\hat{x}_0, \ldots, \hat{x}_{N+1}\}$, for each node $x_k \in X_N \setminus \hat{X}_{N+1}$ the set Ω_j is identified (see Section 3.2.3) such that $(\hat{X}_{N+1} \cup \{x_k\}) \setminus \{\hat{x}_j\}$ is the nodal set of a positive and interpolatory quadrature rule. If there is an x_k such that $\hat{x}_j \notin X_N$, we set $\hat{X}_{N+1} \leftarrow (\hat{X}_{N+1} \cup \{x_k\}) \setminus \{\hat{x}_j\}$ and keep repeating this procedure until no such x_k exists anymore. If there are multiple x_k that could possibly be used to trigger a replacement in $\hat{X}_{N+1}$, the smallest one is selected in the example presented in this article.

The nodes from X_N that cannot be added to $\hat{X}_{N+1}$ are reconsidered in consecutive iterations and added again if possible. It is difficult to theoretically quantify the number of nodes from X_N that can be "added" this way to $\hat{X}_{N+1}$, though it is straightforward to see that there exists at least a single $x_k \in X_N$ that can be reused.

To demonstrate this procedure numerically, let X_1 and W_1 form a Gaussian quadrature rule of two nodes. If the uniform distribution is considered, evaluating all quadrature rules up to $N = 19$ requires in total 22 unique evaluations of u, which is two more than optimally possible considering the limitations of the framework as discussed in this work. The obtained sequence is depicted in Fig. 3(a) (the two additional evaluations of u can be found at $N = 15$ and $N = 18$). This result seems to be somewhat independent from the distribution, since applying the same approach to construct a sequence of



(a) Uniform (b) Beta(10, 10)

Fig. 3. Partially nested, positive, and interpolatory quadrature rules constructed using sequences of Gaussian quadrature rules. The Nth quadrature rule is constructed by iteratively replacing all nodes of a Gaussian quadrature rule of N nodes by the plotted quadrature rule of $N-1$ nodes. The procedure is initiated using the Gaussian quadrature rule consisting of two nodes. The colors indicate the weights of the nodes.

quadrature rules with respect to a Beta(10, 10) distribution requires in total 23 function evaluations, which is three more than optimally possible (the obtained rules are depicted in Fig. 3(b)).

The main advantage of this approach compared to the previously discussed Patterson extension is that it always has positive weights. Moreover the expressions to compute the nodes contained in the quadrature rule are straightforward. However, the approach has the same disadvantage as the removal of nodes (see Section 2.3), since it requires a sequence of existing quadrature rules.

4. Addition of multiple nodes

In the previous section a counterexample of a positive interpolatory quadrature rule is discussed that cannot be extended by adding a single node. In this section we will therefore study the addition of multiple nodes to a quadrature rule. The problem setting is that of Section 2.4: given a positive interpolatory quadrature rule X_N, W_N, determine M as small as possible and nodes X_{N+M} with $X_N \subset X_{N+M}$ such that X_{N+M} forms the nodal set of a positive interpolatory quadrature rule.

The first step is to extend the derivation of Section 3.1 for the addition of multiple nodes. The derivation is again based on Cramer's rule. With the theory that is derived in upcoming Section 4.1 it is not obvious how nodes can be added to the quadrature rule, but it provides geometrical insight in the location of such nodes with respect to the existing nodes. Again we can derive some non-trivial adjustments one can apply to a quadrature rule. These are discussed in Section 4.2. Similar to the case of a single node, there is a tight relation with the Patterson extension. In this case, the Patterson extension for general M is recovered. This is discussed in Section 4.3, including some examples of nested quadrature rules obtained with the theory derived in this section.

4.1. Positive weight criterion

The idea is similar to the derivation of the addition of single node. Let X_N be the initial nodal set and let M be given. The goal is to determine X_{N+M} with $X_N \subset X_{N+M}$ such that it forms the nodal set of a positive interpolatory quadrature rule.

Let $w_k^{(N)}$ for $k = 0, \ldots, N$ be the weights of W_N and likewise let $w_k^{(N+M)}$ be the (unknown) weights of W_{N+M}. Then there exists a vector $\mathbf{c} = (c_0, \ldots, c_N, c_{N+1}, \ldots, c_{N+M})^{\mathrm{T}}$ such that $w_k^{(N+M)} = w_k^{(N)} + c_k$. The goal is to construct $\mathbf{c}$ such that the obtained rule is interpolatory and positive.

With a similar reasoning as before it is straightforward to observe that the following should hold for such a vector to ensure that the obtained quadrature rule is interpolatory:

$$\sum_{k=0}^{N+M} x_k^j c_k = 0, \quad \text{for } j = 0, \ldots, N,$$

and

$$\sum_{k=0}^{N+M} x_k^j c_k = \varepsilon_j, \quad \text{for } j = N+1, \ldots, N+M,$$

where ε_j is as previously introduced, i.e. $\varepsilon_j := \mu_j - \sum_{k=0}^{N} x_k^j w_k^{(N)}$. This can be written in the form of a linear system as follows:

$$\begin{pmatrix} x_0^0 & \cdots & x_N^0 & x_{N+1}^0 & \cdots & x_{N+M}^0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_0^N & \cdots & x_N^N & x_{N+1}^N & \cdots & x_{N+M}^N \\ x_0^{N+1} & \cdots & x_N^{N+1} & x_{N+1}^{N+1} & \cdots & x_{N+M}^{N+1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_0^{N+M} & \cdots & x_N^{N+M} & x_{N+1}^{N+M} & \cdots & x_{N+M}^{N+M} \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_N \\ c_{N+1} \\ \vdots \\ c_{N+M} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \varepsilon_{N+1} \\ \vdots \\ \varepsilon_{N+M} \end{pmatrix}.$$

Applying Cramer's rule to this system requires more bookkeeping, as the right hand side contains multiple non-zero entries. Let $\boldsymbol{\varepsilon} = (0, \ldots, 0, \varepsilon_{N+1}, \ldots, \varepsilon_{N+M})^{\mathrm{T}}$, then Cramer's rule prescribes

$$c_k = \frac{\det V_k(X_{N+M})}{\det V(X_{N+M})},$$

where $V_k(X_{N+M})$ is equal to $V(X_{N+M})$ with the kth column (indexed from 0) replaced by $\boldsymbol{\varepsilon}$. The numerator can be further expanded as follows:

$$\det V_k(X_{N+M}) = \sum_{j=N+1}^{N+M} (-1)^{(j+1)+(k+1)} \varepsilon_j \det V_{(j,k)}(X_{N+M}) = \sum_{j=N+1}^{N+M} (-1)^{j+k} \varepsilon_j \det V_{(j,k)}(X_{N+M}),$$

where $V_{(j,k)}(X_{N+M})$ is the (j,k)-minor of $V(X_{N+M})$ (i.e. the matrix without its jth row and kth column, where both indices start at 0). Hence for c_k the following expression is obtained:

$$\begin{aligned} c_k &= \sum_{j=N+1}^{N+M} (-1)^{j+k} \varepsilon_j \frac{\det V_{(j,k)}(X_{N+M})}{\det V(X_{N+M})} \\ &= \sum_{j=N+1}^{N+M} (-1)^{N+M-j} \varepsilon_j \frac{\det V_{(j,k)}(X_{N+M})}{\det V_{(N+M,k)}(X_{N+M})} \Bigg/ \left(\prod_{\substack{j=0 \\ j \neq k}}^{N+M} (x_k - x_j) \right). \end{aligned}$$

The same derivation is commonly used to derive the determinant of a Vandermonde matrix [30,31], and it is well-known that the ratio of determinants obtained in this expression is an elementary symmetric polynomial. The kth elementary symmetric polynomial is generally defined as the sum of all monomial permutations of length k, that is as follows:

$$e_k(x_0, \ldots, x_N) = \sum_{0 \leq i_1 < \cdots < i_k \leq N} x_{i_1} \cdots x_{i_k}.$$

The elementary symmetric polynomials are only defined for $k \leq N+1$ and by convention $e_0 \equiv 1$. Concluding, the following expression is obtained for c_k:

$$c_k = \left(\sum_{j=N+1}^{N+M} (-1)^{N+M-j} \varepsilon_j e_{N+M-j}(X_{N+M} \setminus \{x_k\}) \right) \Bigg/ \left(\prod_{\substack{j=0 \\ j \neq k}}^{N+M} (x_k - x_j) \right), \quad \text{for } k = 0, \ldots, N+M.$$

Here, e_k is the kth elementary symmetric polynomial as defined above. With a little abuse of notation, we used:

$$\begin{aligned} e_{N+M-j}(X_{N+M} \setminus \{x_k\}) &:= e_{N+M-j}(x_0, \ldots, x_{k-1}, 0, x_{k+1}, \ldots, x_{N+M}) \\ &= e_{N+M-j}(x_0, \ldots, x_{k-1}, x_{k+1}, \ldots, x_{N+M}). \end{aligned}$$

We are now in a position to formulate a theorem in similar form as Theorem 2, but then for multiple nodes. The proof is omitted, since it is equivalent to that of Theorem 2, but then with the equalities derived in this section.

Theorem 7. *Let X_N, W_N form an interpolatory quadrature rule. Then $X_{N+M} = X_N \cup \{x_{N+1}, \ldots, x_{N+M}\}$ forms the nodal set of a positive interpolatory quadrature rule if and only if*

$$-\left(\sum_{j=N+1}^{N+M} (-1)^{N+M-j} \varepsilon_j e_{N+M-j}(X_{N+M} \setminus \{x_k\}) \right) \Bigg/ \left(\prod_{\substack{j=0 \\ j \neq k}}^{N+M} (x_k - x_j) \right) \leq w_k^{(N)}, \quad \text{for } k = 0, \ldots, N+M.$$

For $M = 1$, we have that the summation only incorporates $j = N+1$, hence $e_{N+M-j}(X_{N+M} \setminus \{x_k\}) = e_0(X_{N+M} \setminus \{x_k\}) = 1$, recovering Theorem 2. So Theorem 7 is indeed a strict generalization of Theorem 2.

4.2. Quadrature rule adjustments

Theorem 7 presents a necessary and sufficient condition for a quadrature rule extended with M nodes to have positive weights. Contrary to the addition of a single node, it cannot be used directly to determine possible nodes that can be added to the quadrature rule. This can be seen by rewriting it in a similar form as (3.7), i.e. for $k = 0, \dots, N+M$:

$$\begin{aligned} w_k^{(N)} \prod_{\substack{j=0 \\ j\neq k}}^{N+M} (x_k - x_j) &\geq -\sum_{j=N+1}^{N+M} (-1)^{N+M-j} \varepsilon_j e_{N+M-j}(X_{N+M} \setminus \{x_k\}) \quad \text{if} \quad \prod_{\substack{j=0 \\ j\neq k}}^{N+M} (x_k - x_j) \geq 0, \\ w_k^{(N)} \prod_{\substack{j=0 \\ j\neq k}}^{N+M} (x_k - x_j) &\leq -\sum_{j=N+1}^{N+M} (-1)^{N+M-j} \varepsilon_j e_{N+M-j}(X_{N+M} \setminus \{x_k\}) \quad \text{if} \quad \prod_{\substack{j=0 \\ j\neq k}}^{N+M} (x_k - x_j) \leq 0. \end{aligned} \tag{4.1}$$

Notice that, if $x_{N+1}, \dots, x_{N+M}$ are unknowns, an M-variate system of $N+M+1$ polynomial inequalities is obtained. In general these systems are very difficult to solve, so we do not directly pursue a solution of the system above. Nonetheless, the system still provides a geometrical interpretation about where solutions reside, similar to the case of single node addition (though less intuitive). This is discussed in Section 4.2.1. Based on these geometrical insights, procedures to replace nodes and to add nodes, which extend those explained previously, can be derived. These procedures are discussed in Sections 4.2.2 and 4.2.3 respectively.

4.2.1. Geometry of nodal addition

The type of the inequalities (4.1) (i.e. "greater than" versus "less than") does not change between two nodes and if this type is fixed, the system consists of polynomial inequalities. Hence the region where M nodes can be added is described by a continuous boundary, bounded by the polynomial inequalities of (4.1), consisting of lines, surfaces, or "hypersurfaces" through the nodes.

If one of the right hand sides of (4.1) changes sign, there is an addition of M nodes such that the inequality forms an equality for a specific k. In such cases, there is an addition such that one of the nodes obtains a weight equal to zero. This is equivalent to the case discussed in Section 3.2.3, where a single node is added in order to set the weights of another node equal to zero.

It is difficult to visualize the addition of M nodes in a similar way as we visualized the addition of one node, as there are M nodes $x_{N+1}, \dots, x_{N+M}$ and M quadrature rule errors $\varepsilon_{N+1}, \dots, \varepsilon_{N+M}$. Plotting the errors with respect to the nodes (as in Fig. 2) is therefore not viable, as this is a plot from $\mathbb{R}^M$ to $\mathbb{R}^M$.

On the other hand, if the distribution $\rho(x)$ is fixed beforehand, the values of $\varepsilon_{N+1}, \dots, \varepsilon_{N+M}$ are known and contour plots of the regions encompassing all M nodes that can be added can be made (provided that M is small enough).

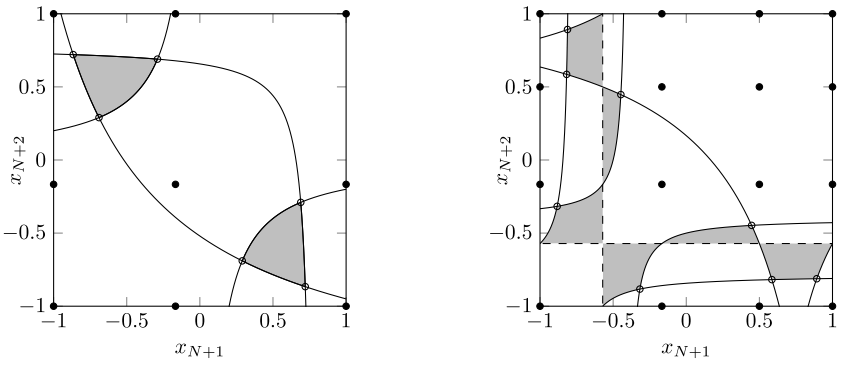
Example 6. Let $\rho \equiv 1/2$ with $\Omega = [-1, 1]$ and reconsider the quadrature rule from Example 1. In Fig. 4 lines are depicted where the inequalities from (4.1) are equalities. The shaded area depicts regions where all inequalities are valid, i.e. any coordinate (x_{N+1}, x_{N+2}) in the shaded region can be added to the respective quadrature rule in order to obtain a positive interpolatory rule. The figure is obviously symmetric around $x_{N+1} = x_{N+2}$, as the order of addition (i.e. first adding x_{N+1} and then x_{N+2} or vice versa) yields equivalent quadrature rules. Selecting a coordinate (x_{N+1}, x_{N+2}) on one of the boundaries results in one weight equal to zero. Adding the coordinates on the corners, depicted by the open circles (i.e. "the boundary of the boundary"), results in two weights equal to zero.

The dashed lines indicate where the inequalities (4.1) with $k = N+1$ and $k = N+2$ change sign. If this happens, one of the new nodes x_{N+1} or x_{N+2} has weight equal to zero. This line forms everywhere a boundary of the shaded area: the node with weight equal to zero can be replaced by any other node, while still resulting in an interpolatory quadrature rule with positive weights. This situation is equivalent to adding a single node x_{N+1} to the quadrature rule, but gaining two degrees, as discussed in Section 3.2.2.

The addition and replacement of multiple nodes follow readily from this example. Notice that if any coordinate $(x_{N+1}, \dots, x_{N+M})$ is known, the replacement for $M = 1$ can be used to reach any other coordinate $(x_{N+1}, \dots, x_{N+M})$ in the same region (shaded in Fig. 4). Hence if all corners of those regions are determined (depicted as open circles in Fig. 4), the full region can be explored straightforwardly using Algorithm 3. As these corner cases form a replacement of nodes, we start by discussing replacement of M nodes. Moreover, it will be shown that these corners are a Patterson extension. Based on the algorithm to determine all these corners, addition of M nodes follows straightforwardly.

4.2.2. Replacement of multiple nodes

Let X_N, W_N be an interpolatory quadrature rule and let indices $k_1, \dots, k_M$ be given such that $0 \leq k_i \leq N$ and $k_i \neq k_j$ for $i \neq j$. In this section the goal is to determine the interpolatory quadrature rule X_{N+M}, W_{N+M} such that $w_{k_i}^{(N+M)} = 0$ for all k_i. Notice that this is equivalent to replacing the nodes $x_{k_1}, \dots, x_{k_M}$ in the quadrature rule X_N by the nodes $x_{N+1}, \dots, x_{N+M}$. The nodes with this property are the intersections of the polynomials of (4.1) and they are depicted as open circles in Fig. 4. Moreover, they describe the boundary of the set of nodes that can be added to the quadrature rule.



(a) $X_N = \{-1, -1/6, 1\}$ (b) $X_N = \{-1, -1/6, 1/2, 1\}$

Fig. 4. Two examples of addition of two nodes to a quadrature rule. In both cases, $\rho \equiv 1/2$. Choosing the two nodes in a shaded area yields positive weights. Choosing the two nodes on the open circles yields two weights equal to zero and positive weights. Dashed lines correspond to a zero weight for x_{N+1} or x_{N+2}, i.e. adding x_{N+1} exactly at the dashed line yields a quadrature rule of $N+2$ nodes with degree $N+2$, making the addition of x_{N+2} trivial.

The desired nodes $x_{N+1}, \ldots, x_{N+M}$ can be determined by calculating the Patterson extension of the interpolatory quadrature rule with the nodes $X_N \setminus \{x_{k_1}, \ldots, x_{k_M}\}$, for which efficient techniques exist [5,11,29]. Such techniques require that M must be known a priori and they do not provide a simple geometrical interpretation. Therefore we proceed by embedding the Patterson extension in the framework discussed here. This yields an alternative, new algorithm to determine these nodes, which is mainly of theoretical and geometrical interest, since it requires the computation of large numbers of roots of polynomials.

We start by solving a slightly easier problem. Assume $\varepsilon_{N+1} = \cdots = \varepsilon_{N+M-1} = 0$ and $\varepsilon_{N+M} \neq 0$. Notice that, if ε_{N+M} is neglected, any addition of $M-1$ nodes yields a valid quadrature rule (as these nodes have zero weight). Geometrically, a fully shaded figure (if drawn as Fig. 4) is obtained. This can be exploited to determine the desired nodes, as only the value of ε_{N+M} imposes a condition on the nodes $x_{N+1}, \ldots, x_{N+M}$.

The nodes that yield $w_{k_1}^{(N+M)} = \cdots = w_{k_M}^{(N+M)} = 0$ can be found by applying Theorem 7 with $c_{k_i} = -w_{k_i}^{(N)}$ for all i or by consecutively applying Theorem 2. In both cases, the following is obtained:

$$\varepsilon_{N+M} = -w_{k_i}^{(N)} \left(\prod_{\substack{j=0 \\ j \neq k_i}}^{N} (x_{k_i} - x_j) \right) \left(\prod_{j=N+1}^{N+M} (x_{k_i} - x_j) \right), \text{ for } i = 1, \ldots, M. \tag{4.2}$$

In principle this system of polynomial equalities is difficult to solve, but it has a certain structure that can be exploited. To see this, let $\hat{\ell}_M(x)$ be the nodal polynomial of the nodes $x_{N+1}, \ldots, x_{N+M}$:

$$\hat{\ell}_M(x) = \prod_{j=N+1}^{N+M} (x - x_j),$$

which translates the system above to

$$\varepsilon_{N+M} = -w_{k_i}^{(N)} \left(\prod_{\substack{j=0 \\ j \neq k_i}}^{N} (x_{k_i} - x_j) \right) \hat{\ell}_M(x_{k_i}), \text{ for } i = 1, \ldots, M. \tag{4.3}$$

If the nodal polynomial $\hat{\ell}_M$ is known, its roots equal $x_{N+1}, \ldots, x_{N+M}$. The nodal polynomial has degree M and it is known that its leading order coefficient equals 1. Therefore it is useful to introduce the polynomial $q_M(x) := \hat{\ell}_M(x) - x^M$, which has degree $M-1$. Then (4.3) can be rewritten as follows:

$$q_M(x_{k_i}) = \hat{\ell}_M(x_{k_i}) - x_{k_i}^M = -\varepsilon_{N+M} \Bigg/ \left(w_{k_i}^{(N)} \prod_{\substack{j=0 \\ j \neq k_i}}^{N} (x_{k_i} - x_j) \right) - x_{k_i}^M, \text{ for } i = 1, \ldots, M. \tag{4.4}$$

These are M values of a polynomial of degree $M-1$, which is a well-known interpolation problem and can be solved with various well-known methods (such as barycentric interpolation [28]). If q_M is determined, the roots of the polynomial $\hat{\ell}_M(x) = q_M(x) + x^M$ are the nodes $x_{N+1}, \ldots, x_{N+M}$. By construction these nodes are such that $w_{k_i}^{(N+M)} = 0$ for $i = 1, \ldots, M$.

Algorithm 4 Determining X_{N+M} with zero weights

Input: Interpolatory quadrature rule X_N, W_N, indices $k_1, \ldots, k_M$.
Output: Interpolatory quadrature rule X_{N+M}, W_{N+M} such that $w_{k_i}^{(N+M)} = 0$ for all i.

1: $m \leftarrow 1$
2: **for** $k = k_1, \ldots, k_M$ **do**
3: Determine $\hat{\ell}_m$ such that $\hat{\ell}_m(x) = x^m + q_m(x)$ (see (4.4)) and

$$\varepsilon_{N+m} = -w_l \hat{\ell}_m(x) \prod_{\substack{j=0 \\ j \neq l}}^{N+m} (x_l - x_j) \text{ for both } l = k \text{ and } l = N+1, \ldots, N+m-1$$

4: Let $r_1, \ldots, r_m$ be the roots of $\hat{\ell}_m$, i.e. $\hat{\ell}_m(r_k) = 0$
5: $X_{N+m} \leftarrow X_N \cup \{r_1, \ldots, r_m\}$ and determine W_{N+m}
6: $m \leftarrow m + 1$
7: **end for**
8: **Return** X_{N+M}, W_{N+M}

Even though assuming $\varepsilon_{N+1} = \cdots = \varepsilon_{N+M-1} = 0$ is not realistic in practical cases, this procedure can readily be extended to the general case. For this we reuse the replacement step from Section 3.2.3. If $\varepsilon_{N+1} \neq 0$, then a single node is added to the quadrature rule such that $w_{k_1}^{(N+1)} = 0$. This is equivalent to applying Algorithm 2 with $x_{N+1} = x_{(k,l)}$, as discussed in Section 3.2.3. Then the obtained quadrature rule $X_{N+1} \setminus \{x_{k_1}\}$ has $\varepsilon_{N+1} = 0$. By applying the procedure discussed above to these $N+1$ nodes, the nodes x_{N+2} and x_{N+3} can be determined such that $w_{k_2}^{(N+2)} = 0$ and $w_{N+1}^{(N+2)} = 0$, i.e. we enforce that the weight of x_{k_2} is zero and the weight of the previously added node becomes zero. The obtained rule has $N+3$ nodes, where two nodes have weight equal to zero. This is again a replacement, but here *two* nodes get weight equal to zero, which is a generalization of the replacement discussed in Section 3.2.3. Those nodes are removed to reobtain a quadrature rule of $N+1$ nodes and this process is repeated iteratively until X_{N+M} is obtained. The obtained rule can be interpreted as a replacement of M nodes, and yields the open circles from Fig. 4. It is an iterative description: a replacement of M nodes is determined using a replacement of $M-1$ nodes. Geometrically, we iterate over the dimension of the figure and iteratively determine a set of nodes that can be used as a replacement.

The obtained nodes form by definition a Patterson extension of the nodal set $X_N \setminus \{x_{k_1}, \ldots, x_{k_M}\}$, since it holds that $(X_N \setminus \{x_{k_1}, \ldots, x_{k_M}\}) \cup \{x_{N+1}, \ldots, x_{N+M}\}$ has degree $N+M$. The existence of such a Patterson extension is directly coupled to the existence of M nodes that can possibly be added to X_N in the hope of obtaining an interpolatory quadrature rule with positive weights: if M nodes can be added to the quadrature rule, the Patterson extension has positive weights, since it forms the boundary of the set that describes all additions. Moreover, if all Patterson extensions of all sets $X_N \setminus \{x_{k_1}, \ldots, x_{k_M}\}$ for any sequences $(k_1, \ldots, k_M)$ have negative weights or are not real-valued, no addition of M nodes exists.

Hence we have proved the following lemma.

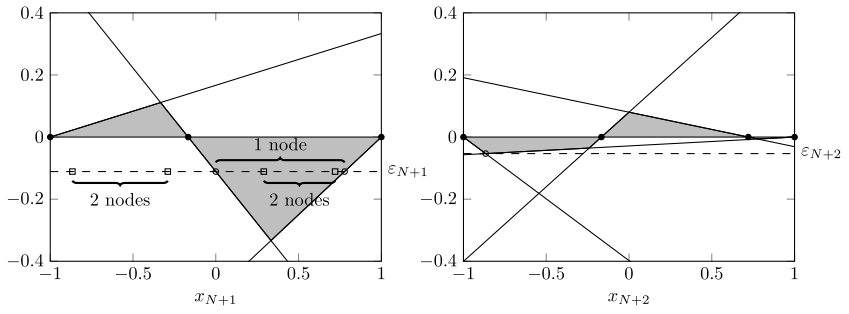
Lemma 8. *Let X_N, W_N form a positive interpolatory quadrature rule, let ρ be the density function (or let a sequence of moments be given) , and let M be given. Then the following statements are equivalent:*

1. *There exists a Patterson extension of M nodes of the quadrature rule X_N, W_M with solely non-negative weights;*
2. *There exist M nodes $x_{N+1}, \ldots, x_{N+M}$ such that $X_N \cup \{x_{N+1}, \ldots, x_{N+M}\}$ forms the nodal set of a positive interpolatory quadrature rule.*

As stated before, any algorithm that computes Patterson extensions can be used to verify whether M nodes exist that can be added to the rule. If a Patterson extension with non-negative weights is found, say $x_{N+1}, \ldots, x_{N+M}$, Algorithm 3 can be used to explore all possible additions to the quadrature rule.

The algorithm based on the geometrical interpretation used in this article is outlined in Algorithm 4. By iterating over all possible sorted sequences $(k_1, \ldots, k_M)$, this procedure can be used straightforwardly to verify whether there exist M nodes that can be added to a given quadrature rule (though this is a costly procedure).

There are two special cases that are (for sake of simplicity) not incorporated in Algorithm 4. Firstly, if $w_k^{(N+m)} = 0$ at the start of an iteration, the polynomial $\hat{\ell}_M(x)$ is not well-defined. This can be incorporated by selecting any non-zero $w_{k_i}^{(N+m)}$ at the start of the iteration. If no such $w_{k_i}^{(N+m)}$ exists, then all these weights are zero, which is the primary goal of the algorithm. Secondly, if $r_k \in X_N$ or $\varepsilon_{N+m} = 0$, a quadrature rule is obtained that has higher degree than its number of nodes. This can be incorporated by combining all double nodes in X_N and likewise adding the respective weights and by skipping any iteration that has $\varepsilon_{N+m} = 0$.



(a) Addition of two nodes (b) Addition of one node

Fig. 5. The addition of 2 nodes to the interpolatory quadrature rule with the nodes $X_N = \{-1, -1/6, 1\}$. *Left:* intervals depicting which node to select if the goal is to add one or two nodes to the quadrature rule. Hence selecting any node between the two squares and adding it yields a quadrature rule to which again a node can be added. The interval of adding a single node is the same as depicted in Fig. 2(a). *Right:* The quadrature rule obtained by adding the rightmost highlighted node of the left figure (i.e. "the rightmost square"). Hence there is only a single node (denoted by the open circle) that can be added to the rule.

4.2.3. Addition of multiple nodes

By combining the quadrature rule replacement of Section 3.2.3 (for $M = 1$) and the replacement of the previous section (for $M > 1$), we obtained a naive algorithm to firstly determine M as small as possible such that there exists a positive interpolatory quadrature rule X_{N+M} (Algorithm 4) and secondly to explore *all* such M nodes (Algorithm 3, yielding the shaded areas of Fig. 4).

Determining the number of nodes M that can be added to an interpolatory quadrature rule can straightforwardly be done by solving (4.2) for each sequence of $k_1, \ldots, k_M$ with $k_1 < \cdots < k_M$. This gives all locations where M nodes have zero weight. If at any of these locations all nodes have non-negative weight, then M nodes can be added to the rule. Otherwise, M is increased and the process is repeated.

Often the value of M is unknown a priori. Besides determining the M nodes that can be added, the goal is also to determine M as small as possible (this is also how we formulated the problem originally in Section 2.4). Algorithm 4 can be used to determine M, as results from previous iterations can be reused. To see this, suppose a quadrature rule is given and by applying Algorithm 4 it is known that no addition of at most $M-1$ nodes exist. Then during these calculations, all sequences of nodes have been determined that make $M-1$ weights zero. By initializing Algorithm 4 with these sequences, only the last iteration of the loop is necessary, which significantly reduces the computational expense.

It is required to repeatedly determine large numbers of polynomial roots in this algorithm. This is nearly impossible to do symbolically, except for some special cases (e.g. $M \leq 3$ or symmetric quadrature rules). Moreover determining the roots numerically can result in quick aggregation of numerical errors. We use variable precision arithmetic, i.e. determine the roots with a large number of significant digits.

For large N this is a costly algorithm, as the number of sorted sequences of length M equals

$$\#(k_1, \ldots, k_M) = \binom{N+1+M}{M},$$

which grows fast for large N. Therefore using this algorithm to compute all removals is slower than using existing techniques to compute the Patterson extension, albeit that it is able to reuse all additions of $M-1$ nodes to compute all additions of M nodes.

If all sets of M nodes have been determined that can be added to the quadrature rule, the techniques from Section 3.2.3 can be used to fully explore all nodes that can be added to the rule. This requires solving linear equalities, which can be done fast and accurately.

The possibility of adding M nodes to the quadrature rule does not guarantee the possibility of adding $M+1$ nodes to the quadrature rule, which is shown in the following example.

Example 7. We revisit the quadrature rule example from Example 1, i.e.

$$X_N = \left\{-1, -\frac{1}{6}, 1\right\}, W_N = \left\{\frac{1}{10}, \frac{24}{35}, \frac{3}{14}\right\}.$$

In Fig. 5(a) regions are depicted where a single node can be added (similar to Fig. 2(a)) and regions where, upon adding a node from that region, another node can be added (this is the projection of Fig. 4(a)). The addition of the rightmost node with the latter property is depicted in Fig. 5(b), demonstrating that there is a single node that can be added and that this is indeed a limiting case.

Notice that the intervals where a single node and where two nodes can be added are independent from each other. There exist pairs of nodes x_{N+1}, x_{N+2} firstly such that both W_{N+1} and W_{N+2} are all positive (in the right interval surrounded by squares), secondly such that W_{N+1} is positive, but W_{N+2} is not (the right interval surrounded by circles, outside the interval surrounded by squares), thirdly such that W_{N+1} is not positive, but W_{N+2} is (the left interval surrounded by squares), and finally such that both W_{N+1} and W_{N+2} are always negative (outside all intervals).

4.3. Constructing quadrature rules

Similar to the case of addition of a single node, the Patterson extension is obtained for specific choices of nodes that are added to the rule. In fact, the nodes determined with Algorithm 4 are a Patterson extension of a quadrature rule with a smaller number of nodes. As the Gaussian quadrature rule is a special case of the Patterson extension, this rule also follows from the framework discussed in this article. This is discussed in more detail in Section 4.3.1.

By repeatedly applying Algorithm 4, a sequence of nested quadrature rules can be determined. These rules and their properties are considered in Section 4.3.2.

4.3.1. Patterson extension

The boundary of the set that describes all possible additions is spanned by the Patterson extension (the open circles in Figs. 2(a) and 4(a)). These nodes have the property that, upon adding them to the quadrature rule, a rule of degree $N+M$ is obtained with M weights equal to zero. This is equivalent to the Patterson extension of the quadrature rule *without* those M nodes with zero weight. For $M = 1$, this was demonstrated in Section 3.3.1.

For general M, the Patterson extension can be deduced mathematically as follows. Let X_N, W_N be a quadrature rule and, as before, let $x_{N+1}, \ldots, x_{N+M}$ be such that the following nodes form a quadrature rule of degree $N+M$:

$$(X_N \cup \{x_{N+1}, \ldots, x_{N+M}\}) \setminus \{x_{k_1}, \ldots, x_{k_M}\}. \tag{4.5}$$

Furthermore, let X_{N-M} be the nodes of an interpolatory quadrature rule of degree $N-M$ be as follows:

$$X_{N-M} = X_N \setminus \{x_{k_1}, \ldots, x_{k_M}\}.$$

Upon adding $\{x_{N+1}, \ldots, x_{N+M}\}$ to X_{N-M}, the nodes from (4.5) are obtained, that have degree $N+M$. Hence M nodes are added to an interpolatory rule of degree $N-M$ and the obtained degree is $N+M$, which is by definition a Patterson extension. Notice that the obtained quadrature rule is interpolatory, but not necessarily positive.

The Gaussian quadrature rule can be deduced as a special case from Algorithm 4. To see this, suppose $M = N+1$, which is the number of nodes of the rule under consideration. In that case, there is only a single sequence of $k_1, \ldots, k_M$, defined as follows up to a permutation:

$$k_j = j - 1 \text{ for } j = 1, \ldots, N+1.$$

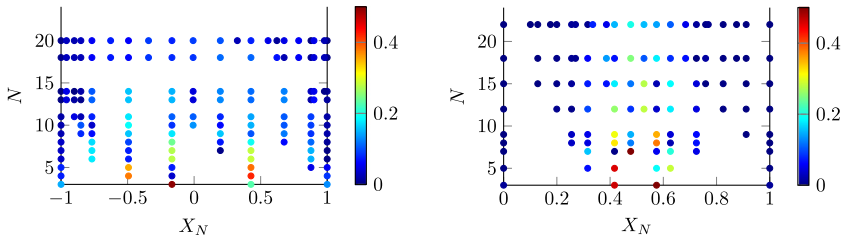
By applying Algorithm 4, the nodes from (4.5) are obtained with $M = N+1$, which are:

$$(X_N \cup \{x_{N+1}, \ldots, x_{2N+1}\}) \setminus \{x_0, \ldots, x_N\} = \{x_{N+1}, \ldots, x_{2N+1}\}.$$

Hence the $N+1$ nodes $x_{N+1}, \ldots, x_{2N+1}$ form a quadrature rule of degree $2N+1$, which is by definition the Gaussian quadrature rule. In other words, when adding a Gaussian quadrature rule to an existing quadrature rule and setting all existing weights to zero, a valid addition is obtained.

Example 8. To demonstrate where Patterson extensions occur in our work, reconsider the interpolatory quadrature rule with the nodes $X_N = \{-1, -1/6, 1\}$. In Section 3.3.1 three different Patterson extensions related to this quadrature rule were discussed: $\{-1/3, 1\}$, $\{-1/6, 2\}$, or $\{-1, 1/3\}$. All these rules are Patterson extensions (of smaller quadrature rules) with $M = 1$. To obtain a Patterson extension with $M = 2$ and subsequently a Gaussian quadrature rule, consider Algorithm 4 using $\{k_1, k_2, k_3\} = \{0, 1, 2\}$. The algorithm proceeds as follows:

1. In the first iteration, it follows that $\hat{\ell}_1(x) = x + 5/3$ and therefore the following quadrature rule is obtained:
$$X_{N+1} = \left\{-1, -\frac{1}{6}, 1, -\frac{5}{3}\right\}, W_{N+1} = \left\{0, \frac{16}{21}, \frac{11}{56}, \frac{1}{24}\right\}.$$
Notice that the node $x_{N+1} = -5/3$ was obtained in Section 3.2.2, where we discussed that after adding this node one obtains $w_0^{(3)} = 0$.
2. In the second iteration, it follows that $\hat{\ell}_2(x) = x^2 + 2/5x - 1/5$. Here, the Patterson extension with $M = 2$ of the quadrature rule with "nodes" $\{1\}$ is obtained. Hence the following rule is obtained (notice that the node $-5/3$ is removed):
$$X_{N+2} = \left\{-1, -\frac{1}{6}, 1, \frac{1}{5}\left(-1-\sqrt{6}\right), \frac{1}{5}\left(-1+\sqrt{6}\right)\right\}, W_{N+2} = \left\{0, 0, \frac{1}{9}, \frac{1}{36}\left(16+\sqrt{6}\right), \frac{1}{36}\left(16-\sqrt{6}\right)\right\}.$$



(a) Uniform (b) Beta(10, 10)

Fig. 6. Nested, positive, and interpolatory quadrature rules, initialized with $X_N = \{-1, -1/6, 1\}$ (left) or $X_N = \{0, 5/12, 1\}$ (right). Given the Nth quadrature rule, the next rule is obtained by firstly computing the minimal number of nodes that can be added and by secondly randomly adding such a minimal number of nodes to the rule. The colors indicate the weights of the nodes.

3. In the third iteration, it follows that $\hat{\ell}_3(x) = x^3 - 3/5x$, whose roots are the Gaussian quadrature rule or, equivalently, the Patterson extension with $M = 3$ of the empty quadrature rule:

$$X_{N+3} = \left\{-1, -\frac{1}{6}, 1, -\frac{1}{5}\sqrt{15}, 0, \frac{1}{5}\sqrt{15}\right\}, W_{N+3} = \left\{0, 0, 0, \frac{5}{18}, \frac{4}{9}, \frac{5}{18}\right\}.$$

In this specific example it is possible to determine all nodes symbolically, but for larger values of M this is generally not possible.

Considering the nodes in a different order results in different intermediate Patterson extensions, but obviously the Gaussian quadrature rule is the rule that is finally obtained. These steps also demonstrate the possibility to store intermediate results: only the nodes of step 2 are necessary to deduce the nodes of step 3.

Specialized algorithms exist for specific distributions and specific values of N and M to construct Gaussian, Gauss–Kronrod, and Gauss–Patterson quadrature rules [4,5], but it remains a challenging topic to determine the Patterson extension for general non-Gaussian quadrature rules. The algorithm presented in this article is not an alternative for these existing algorithms, but embeds the Patterson extension in the discussed framework and can be used to determine *all* M nodes that can be added to a quadrature rule. If an efficient procedure to determine large numbers of Patterson extensions is available, it can be readily used to determine whether an extension for a specific M exists. By consecutively replacing the new nodes (see Section 3.2.3) all M nodes that can be added can be found.

4.3.2. Nested, positive, and interpolatory quadrature rule

Algorithm 4 provides a straightforward procedure to determine the minimal value of M and the positive interpolatory quadrature rule nodes X_{N+M} such that $X_N \subset X_{N+M}$. The replacement procedure for $M = 1$ of Section 3.2.3 can be used to determine all possible nodes, given M. This is the original goal of the article as outlined in Section 2.4 and examples of such quadrature rules are depicted in Fig. 6. Here, each quadrature rule is iteratively extended with a minimal number of nodes, and the nodes that are added are selected randomly from the set containing all M nodes that can be added. There are two main differences with the quadrature rules obtained in Section 3.3.2, where an existing rule was used as basis for a larger quadrature rule: the rules obtained in this section are fully nested, but do add more than one node between two consecutive rules.

Both figures demonstrate that M varies significantly and does not increase monotonically. This is in line with the conclusions drawn in Section 4.2.3, as shown in Fig. 5. Moreover for almost all N, the value of M is significantly larger in case the Beta distribution is considered, which is related to the "bad" initial set of nodes for this distribution. A different initialization would lead to different values of M.

5. Numerical integration with positive quadrature rules

This article is concerned with the construction of quadrature rules with *positive* weights and two new quadrature rules have been introduced: one based on the consecutive replacement of single nodes (possibly resulting in a sequence of rules that is not nested) and one by randomly adding nodes ensuring positive weights. We briefly assess the numerical performance of these quadrature rules by means of the Genz test functions (see Table 1). The Genz test functions [32] are functions defined on $\Omega = [0, 1]$ constructed specifically to test integration routines. Each function has a specific family attribute that is considered to be challenging for integration routines, that can be enlarged by a shape parameter a and translated by a translation parameter b. We restrict ourselves to the uniform distribution, as in this case the exact value of the integral of the Genz functions is known analytically.

Table 1
The test functions from Genz [32], which depend on the shape and translation parameters a and b.

Integrand family	Attribute
$u_1(x) = \cos(2\pi b + ax)$	Oscillatory
$u_2(x) = \left(a^{-2} + (x-b)^2\right)^{-1}$	Product Peak
$u_3(x) = (1+ax)^{-2}$	Corner Peak
$u_4(x) = \exp\left(-a^2(x-b)^2\right)$	Gaussian
$u_5(x) = \exp(-a\lvert x-b\rvert)$	C^0 function
$u_6(x) = \begin{cases} 0 & \text{if } x > b \\ \exp(ax) & \text{otherwise} \end{cases}$	Discontinuous

We consider the performance of the following four quadrature rules:

1. A quadrature rule that is determined by consecutively adding and replacing nodes originating from a Gaussian quadrature rule (see Fig. 3(a)). This rule was discussed in Section 3.3 and is a partially nested, positive, and interpolatory quadrature rule. The rule is initialized with the quadrature rule nodes $X_N = \{0, 5/12, 1\}$ (i.e. the nodes from the example as discussed before, translated to $[0, 1]$).
2. A quadrature rule that is determined by consecutively randomly adding M nodes to the rule such that the obtained rule is positive. Here M is minimal, i.e. the smallest number of nodes is added for each N (see Fig. 6(a)). This rule was discussed in Section 4.3 and is a nested, positive, and interpolatory quadrature rule. The rule is initialized in the same way as the quadrature rule of the previous point, i.e. using $X_N = \{0, 5/12, 1\}$.
3. The Clenshaw–Curtis quadrature rule [15], where the nodes X_N are defined explicitly by (2.5). It is well known that these nodes have positive weights if the distribution under consideration is uniform, which is the case. This positive and interpolatory quadrature rule is nested for specific levels, i.e. $X_{N_L} \subset X_{N_{L+1}}$ with $N_L = 2^L$ $(l = 1, 2, \dots)$.
4. The Gaussian quadrature rule [4], where the nodes and weights are defined as the quadrature rule with $N+1$ nodes of degree $2N+1$. This quadrature rule is not nested, so refining the quadrature rule results in a significant number of new function evaluations.

The error measure e_N is the absolute integration error, i.e.

$$e_N(u) = |\mathcal{I}u - \mathcal{A}_N u|,$$

where $u = u_g$ with $g = 1, \dots, 6$, i.e. u is one of the Genz test functions. To obtain meaningful results we select the parameters a and b randomly in the unit interval and repeat the experiment 100 times. The errors reported here are averaged over the 100 experiments and are therefore denoted by $\bar{e}_N$.

It is instructive to compare the error with the upper bound that follows from the Lebesgue inequality (2.4):

$$e_N(u) \leq 2 \inf_{\varphi \in \mathbb{P}(N)} \|u - \varphi\|_\infty, \tag{5.1}$$

where we use that $\mu_0 = 1$ in our test cases. This error is determined using the algorithm of Remez [24, Chapter 3], with the implementation from `chebfun` [33]. Convergence results for the uniform distribution $\rho \equiv 1$ in $\Omega = [0, 1]$ are gathered in Fig. 7.

Notice that regardless of the function under consideration all quadrature rule errors remain far under the dashed line, that represents the right-hand side of (5.1). This shows that the bound from this inequality is far from sharp.

The first four Genz functions can be approximated well using polynomials, as they are analytic and have rapidly converging Chebyshev coefficients. The best approximation converges exponentially in these cases, which is also the case for the four quadrature rules under consideration. The quadrature rules determined using the framework of this article perform slightly worse than the Clenshaw–Curtis and the Gaussian quadrature rule. This is related to the fact that these rules exploit the structure of the underlying distribution to a large extent (e.g. symmetry and higher-order moments), whereas the rules in this work only optimize for the positivity of the weights. The Gaussian quadrature rule converges with the highest rate, which is related to its high polynomial degree (a rule of $N+1$ nodes has degree $2N+1$). However, the Gaussian rule is not nested, so to refine the estimate of the integral for increasing number of nodes the number of function evaluations increases significantly. If a computationally expensive function is considered, using a nested quadrature rule with fine granularity (such as the proposed rules) significantly reduces the cost of refining the quadrature rule estimate.

The fifth Genz test function is not differentiable and can therefore not be approximated well using a polynomial. This can be observed from the best approximation polynomial, that converges with order 1 (so we would expect that $\bar{e}_N \sim 1/N$). In this case the difference between the Gaussian rule and the other rules is significantly smaller, demonstrating that the high polynomial degree of Gaussian rules is less relevant if the integrand is not smooth.

The sixth Genz test function cannot be approximated accurately using a polynomial when considering the ∞-norm, as it is discontinuous. Hence the best approximation error remains constant. However, the approximation of the quadrature rules still converges with order 1/2. In this case, there is a clear difference between the integration error (that is an averaged error) and the best approximation error (that is a uniform error).

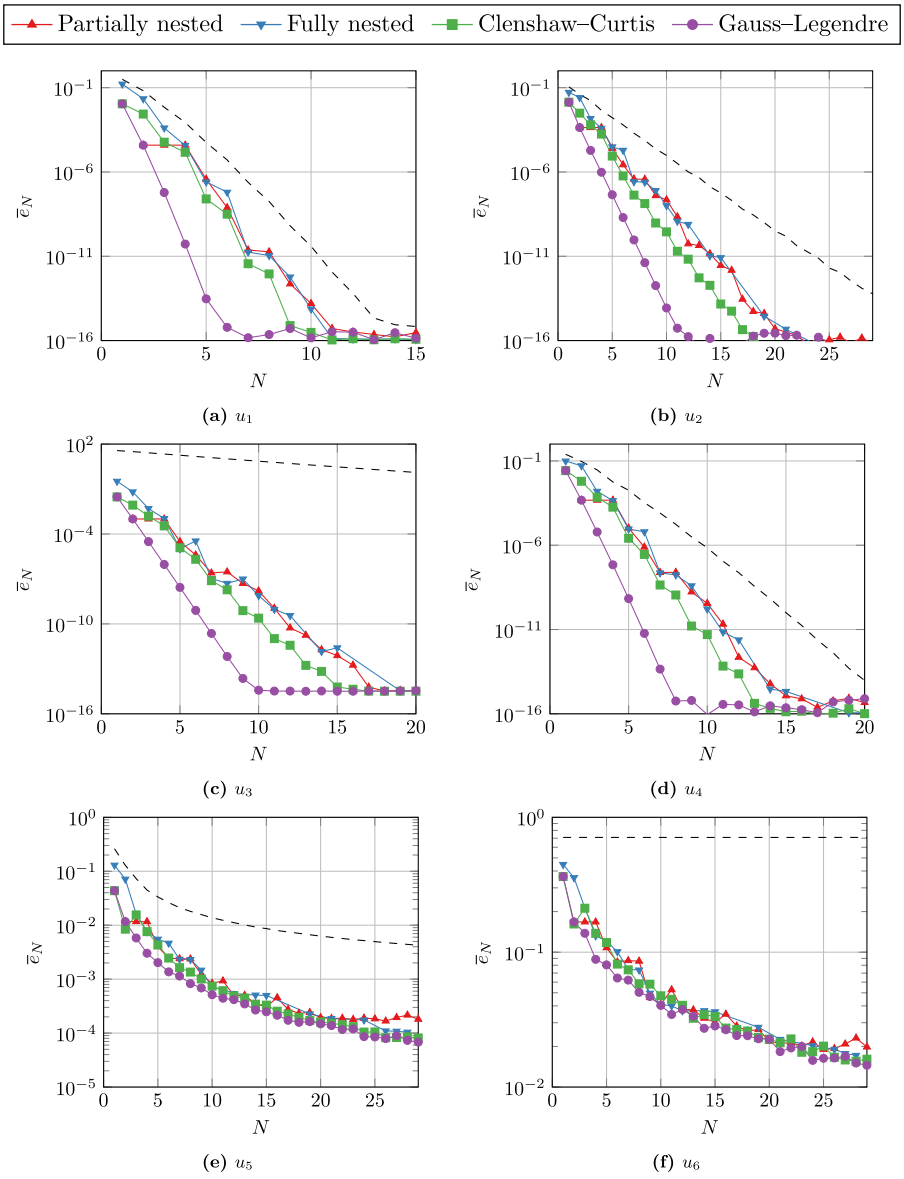


Fig. 7. Convergence of the Genz test functions using various quadrature rule techniques. The absolute error of the best approximation polynomial (i.e. $\inf_{\varphi\in\Phi_N} \|u-\varphi\|_\infty$) is dashed.

6. Conclusion

In this article, a novel mathematical framework is presented for the construction of nested, positive, and interpolatory quadrature rules by using a geometrical interpretation. Given an existing quadrature rule, necessary and sufficient conditions have been derived for M new nodes to form an interpolatory quadrature rule with positive weights. The conditions have been formulated as inequalities, which are explicit if $M = 1$ and implicit if $M > 1$.

The addition of a single node can be treated as a special case, which can be solved analytically. The analytical expression can be used to add nodes to and replace nodes within a quadrature rule. The addition of multiple nodes can be determined numerically and a naive algorithm is presented for this purpose. Based on the quadrature rules obtained by this algorithm, the set that encompasses all additions of M nodes can be explored by iteratively replacing nodes.

The well-known Patterson extension of quadrature rules forms a special case of the framework, as it is obtained by constructing the quadrature rules with M weights equal to zero. As such, our proposed framework and its geometrical interpretation are well embedded in existing theory on the addition of nodes to quadrature rules. The framework

provides various possibilities to construct or adapt quadrature rules and two examples have been discussed: one based on consecutively adding and replacing one node and one based on consecutively adding multiple nodes.

Numerical integration using the two quadrature rules introduced in this work shows the key advantages of nested quadrature rules with positive weights: estimates computed using the quadrature rules are stable and nesting allows for computationally cheap refinements of the estimates. Existing quadrature rules, such as the Gaussian and the Clenshaw–Curtis quadrature rule, are not nested with the fine granularity as the rules in this work.

There are various options to further extend the framework set out in this article. The algorithm to determine whether multiple nodes exist that can be added to the quadrature rule depends on determining many polynomial roots and iterates over all possible sequences of nodes that can become zero. For a large number of nodes this is computationally very costly and therefore warrants the need to derive an efficient algorithm to determine these nodes. Moreover the framework set out in this article does not use the relations that exists between consecutive moments of a distribution [34], which can possibly be used to further extend the framework set out in this text.

Acknowledgments

This research is part of the Dutch EUROS program, which is supported by NWO domain Applied and Engineering Sciences and partly funded by the Dutch Ministry of Economic Affairs, The Netherlands.

References

[1] G. Pólya, Über die Konvergenz von Quadraturverfahren, Math. Z. 37 (1) (1933) 264–286, http://dx.doi.org/10.1007/bf01474574.
[2] R. Cools, Constructing cubature formulae: the science behind the art, Acta Numer. 6 (1997) 1, http://dx.doi.org/10.1017/s0962492900002701.
[3] H. Brass, K. Petras, in: R.L. Cohen, J.S. Ellenberg, M.A. Singer, B. Sudakov (Eds.), Quadrature Theory, in: Mathematical Surveys and Monographs, vol. 178, American Mathematical Society, 2011, http://dx.doi.org/10.1090/surv/178.
[4] G.H. Golub, J.H. Welsch, Calculation of Gauss quadrature rules, Math. Comp. 23 (106) (1969) 221–230, http://dx.doi.org/10.1090/s0025-5718-69-99647-1.
[5] D.P. Laurie, Calculation of Gauss–Kronrod quadrature rules, Math. Comp. 66 (219) (1997) 1133–1146, http://dx.doi.org/10.1090/s0025-5718-97-00861-2.
[6] D. Vladislav, Construction of Gauss–Kronrod–Hermite quadrature and cubature formulas, Studia Univ. Babeş-Bolyai Math. 49 (3) (2004) 111–117.
[7] G. Monegato, An overview of the computational aspects of kronrod quadrature rules, Numer. Algorithms 26 (2) (2001) 173–196, http://dx.doi.org/10.1023/a:1016640617732.
[8] T.N.L. Patterson, The optimum addition of points to quadrature formulae, Math. Comp. 22 (104) (1968) http://dx.doi.org/10.1090/s0025-5718-68-99866-9, 847–847.
[9] D.K. Kahaner, G. Monegato, Nonexistence of extended Gauss–Laguerre and Gauss–Hermite quadrature rules with positive weights, Z. Angew. Math. Phys. 29 (6) (1978) 983–986, http://dx.doi.org/10.1007/bf01590820.
[10] D.K. Kahaner, J. Waldvogel, L.W. Fullerton, Addition of points to Gauss–Laguerre quadrature formulas, SIAM J. Sci. Stat. Comput. 5 (1) (1984) 42–55, http://dx.doi.org/10.1137/0905003.
[11] D.P. Laurie, Anti-Gaussian quadrature formulas, Math. Comp. 65 (214) (1996) 739–748, http://dx.doi.org/10.1090/s0025-5718-96-00713-2.
[12] J. Ma, V. Rokhlin, S. Wandzura, Generalized Gaussian quadrature rules for systems of arbitrary functions, SIAM J. Numer. Anal. 33 (3) (1996) 971–996, http://dx.doi.org/10.1137/0733048.
[13] A. Genz, B.D. Keister, Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight, J. Comput. Appl. Math. 71 (2) (1996) 299–309, http://dx.doi.org/10.1016/0377-0427(95)00232-4.
[14] R. Bourquin, Exhaustive search for higher-order Kronrod–Patterson Extensions, 2015–11, ETH Zürich, 2015.
[15] C.W. Clenshaw, A.R. Curtis, A method for numerical integration on an automatic computer, Numer. Math. 2 (1) (1960) 197–205, http://dx.doi.org/10.1007/bf01386223.
[16] B.A. Ibrahimoglu, Lebesgue functions and lebesgue constants in polynomial interpolation, J. Inequal. Appl. 2016 (1) (2016) 93, http://dx.doi.org/10.1186/s13660-016-1030-3.
[17] L.N. Trefethen, Spectral Methods in MATLAB, Society for Industrial & Applied Mathematics (SIAM), 2000, http://dx.doi.org/10.1137/1.9780898719598.
[18] T. Hasegawa, H. Sugiura, T. Torii, Positivity of the weights of extended Clenshaw–Curtis quadrature rules, Math. Comp. 60 (202) (1993) http://dx.doi.org/10.1090/s0025-5718-1993-1176710-2, 719–719.
[19] L.M.M. van den Bos, B. Koren, R.P. Dwight, Non-intrusive uncertainty quantification using reduced cubature rules, J. Comput. Phys. 332 (2017) 418–445, http://dx.doi.org/10.1016/j.jcp.2016.12.011.
[20] F. Peherstorfer, Characterization of positive quadrature formulas, SIAM J. Math. Anal. 12 (6) (1981) 935–942, http://dx.doi.org/10.1137/0512079.
[21] G. Dahlquist, Å. Björck, Numerical Methods in Scientific Computing: Volume 1, Society for Industrial and Applied Mathematics, 2008.
[22] R. Cools, A. Haegemans, On the construction of multi-dimensional embedded cubature formulae, Numer. Math. 55 (6) (1989) 735–745, http://dx.doi.org/10.1007/bf01389339.
[23] M.W. Wilson, A general algorithm for nonnegative quadrature formulas, Math. Comp. 23 (106) (1969) http://dx.doi.org/10.1090/s0025-5718-1969-0242374-1, 253–253.
[24] G.A. Watson, Approximation Theory and Numerical Methods, John Wiley & Sons Ltd, 1980.
[25] C. Zhou, Convergence of Gaussian quadrature formulas on infinite intervals, J. Approx. Theory 123 (2) (2003) 280–294, http://dx.doi.org/10.1016/s0021-9045(03)00090-x.
[26] P. Rabinowitz, J. Kautsky, S. Elhay, J. Butcher, On Sequences of Imbedded Integration Rules, 86–02, The University of Adelaide, 1986.
[27] L.M.M. van den Bos, B. Sanderse, W.A.A.M. Bierbooms, G.J.W. van Bussel, Generating nested quadrature rules with positive weights based on arbitrary sample sets, SIAM/ASA J. Uncertain. Quantif. 8 (1) (2020) 139–169, http://dx.doi.org/10.1137/18M1213373.
[28] J.-P. Berrut, L.N. Trefethen, Barycentric Lagrange interpolation, SIAM Rev. 46 (3) (2004) 501–517, http://dx.doi.org/10.1137/s0036144502417715.
[29] T.N.L. Patterson, An algorithm for generating interpolatory quadrature rules of the highest degree of precision with preassigned nodes for general weight functions, ACM Trans. Math. Software 15 (2) (1989) 123–136, http://dx.doi.org/10.1145/63522.63523.
[30] W. Gautschi, On inverses of vandermonde and confluent vandermonde matrices, Numer. Math. 4 (1962) 117–123, http://dx.doi.org/10.1007/bf01386302.
[31] N. Macon, A. Spitzbart, Inverses of Vandermonde matrices, Amer. Math. Monthly 65 (2) (1958) 95–100, http://dx.doi.org/10.2307/2308881.

[32] A. Genz, Testing multidimensional integration routines, in: Proceedings of the International Conference on Tools, Methods and Languages for Scientific and Engineering Computation, Elsevier North–Holland, 1984, pp. 81–94.
[33] T.A. Driscoll, N. Hale, L.N. Trefethen, Chebfun Guide, Pafnuty Publications, 2014.
[34] J.A. Shohat, J.D. Tamarkin, The problem of moments, Mathematical Surveys, vol. 1, American Mathematical Society, 1943.