

# A geometric approach for the addition of nodes to an interpolatory quadrature rule with positive weights

L.M.M. van den Bos<sup>\*1,2</sup> and B. Sanderse<sup>1</sup>

<sup>1</sup>*Centrum Wiskunde & Informatica, P.O. Box 94079, 1090 GB, Amsterdam*

<sup>2</sup>*Delft University of Technology, P.O. Box 5, 2600 AA, Delft*

21st February 2019

## Abstract

A novel mathematical framework is derived for the addition of nodes to 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 or replaced are explicit. It is shown that, depending on the location of existing nodes and moments of the distribution, addition of a single node and preservation of positive weights is not always possible. 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. Moreover, in this case the inequalities describing the regions where nodes can be added become implicit. An algorithm is presented to explore these regions and it is shown that the well-known Patterson extension of quadrature rules forms the boundary of these regions. Two new quadrature rules, based on the framework, are proposed and their efficiency is numerically demonstrated by using Genz test functions.

**Keywords:** Quadrature rules (65D32), Numerical integration (65D30), Interpolation (65D05)

## 1 Introduction

This article is concerned with the addition of nodes to an interpolatory quadrature rule 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, interpolatory quadrature rules with positive weights converge among others for any absolute continuous function [4, 6, 27]. Secondly, nested quadrature rules allow for straightforward refinements of the quadrature rule approximation, which is especially relevant if the integrand is computationally expensive.

Possibly the best-known interpolatory quadrature rule is the Gaussian quadrature rule [12], 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 [18, 21]. The Gauss–Kronrod–Patterson quadrature rule [23, 24] 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 [15, 16]. Even though many other extensions have been proposed over the years [11, 17, 19], in general it is difficult to obtain a series of nested quadrature rules with positive weights [22]. Moreover often the smallest possible granularity between two consecutive nested quadrature rules can only be found by exhaustive search [3].

An other large group of well-known quadrature rules is formed by the Clenshaw–Curtis quadrature rules [5], 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 [14], it is well-known that these quadrature rules have positive weights if the distribution under consideration

<sup>\*</sup>Corresponding author: [l.m.m.van.den.bos@cw.i.nl](mailto:l.m.m.van.den.bos@cw.i.nl)

is uniform (explicit expressions are known [29]). Moreover for non-uniform distributions, the condition number of the quadrature rule converges to unity [4]. However, the vanilla Clenshaw–Curtis nodes are only nested for exponentially growing numbers of nodes [13].

Both the Gaussian and Clenshaw–Curtis quadrature rules have explicitly predefined nodes based on the roots of orthogonal polynomials. This results into 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 [2, 7, 26], which yields a necessary and sufficient condition for a quadrature rule to have positive weights. Moreover it further extends upon previous results on the removal of nodes from quadrature rules [2], yielding applicable methods that can be used to 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. We emphasize that the focus of this article is mainly on the mathematical aspects and not on the numerical construction of quadrature rules.

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. Section 5 contains numerical examples of quadrature rules to demonstrate simple applications of the proposed framework. 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 is briefly reviewed in Section 2.2. The problem setting of this article is sketched mathematically in Section 2.3.

### 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 distribution function  $\rho : \Omega \rightarrow \mathbb{R}_+$ . The main interest is to approximate the integral over a given continuous function  $u : \Omega \rightarrow \mathbb{R}$ , i.e. to approximate the following operator:

$$\mathcal{I}u = \int_{\Omega} u(x) \rho(x) dx = \int_a^b u(x) \rho(x) dx.$$

A quadrature rule approximates this integral by means of a weighted average, consisting of nodes and weights, which we denote by  $X_N = \{x_0, \dots, x_N\} \subset \Omega$  and  $W_N = \{w_0, \dots, w_N\} \subset \mathbb{R}$  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  $\rho$  has finite moments, so that is assumed to be the case throughout this article.

The operators  $\mathcal{A}_N$  and  $\mathcal{I}$  and the space  $\mathbb{P}(K)$  are linear, so (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:

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

with  $\mu_k$  the raw moments of  $\rho$ :

$$\mu_k = \int_{\Omega} x^k \rho(x) dx.$$

Throughout this article it is assumed that  $\mu_k$  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

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

such that, given the nodes, (2.2) defines a unique solution of the weights provided that all nodes are distinct. A quadrature rule with distinct nodes that solves the linear system (2.2) (and therefore has degree  $N$ ) is called interpolatory, as it can be formed by integrating the polynomial interpolant of  $u$  using the nodes  $X_N$ .

## 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 [27], which can among others be demonstrated by applying the Lebesgue inequality [4], provided that  $\Omega$  is bounded. To this end, let  $u$  be given and let  $\varphi_N$  be the best approximation polynomial [30] of degree  $N$  of  $u$ , i.e.  $\varphi_N = \arg \min_{\varphi \in \mathbb{P}(N)} \|\varphi - u\|_{\infty}$ . Then

$$\begin{aligned} |\mathcal{A}_N u - \mathcal{I}u| &\leq (\|\mathcal{A}_N\|_{\infty} + \|\mathcal{I}\|_{\infty}) \|u - \varphi_N\|_{\infty} \\ &= (\|\mathcal{A}_N\|_{\infty} + \mu_0) \|u - \varphi_N\|_{\infty}. \end{aligned} \quad (2.4)$$

Here, it holds that

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

The magnitude of  $\|\mathcal{A}_N\|_{\infty}$  scales with the magnitude of  $\mu_0$ , so it is common to measure the accuracy of the quadrature rule by means of its condition number:

$$\kappa_N := \frac{\|\mathcal{A}_N\|_{\infty}}{\mu_0}. \quad (2.5)$$

The condition number is minimal and equal to 1 if and only if all weights are non-negative. Combining (2.4) and (2.5) yields the following inequality:

$$|\mathcal{A}_N u - \mathcal{I}u| \leq (1 + \kappa_N) \mu_0 \inf_{\varphi \in \mathbb{P}(N)} \|u - \varphi\|_{\infty}. \quad (2.6)$$

This shows many similarities with the classical Lebesgue inequality [14, 30] and demonstrates that if  $u$  can be approximated well using a polynomial, it can be integrated using a quadrature rule with small condition number (e.g. a quadrature rule with positive weights). Similar results exist for unbounded  $\Omega$  [4].

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

The Clenshaw–Curtis quadrature rule [5] has nodes  $X_N$  that are defined as follows for  $\Omega = [-1, 1]$ :

$$x_k = \cos\left(\frac{k}{N}\pi\right), \text{ for } k = 0, \dots, N. \quad (2.7)$$

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  $\mu_0$  for large  $N$  [4]. 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 [12] are defined as the roots of the orthogonal polynomials with respect to the distribution  $\rho$  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 [18, 24]. 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  $X_N$ :

$$\int_{\Omega} x^j \left( \prod_{k=0}^{N+M} (x - x_k) \right) \rho(x) dx = 0, \text{ for } j = 0, \dots, M - 1. \quad (2.8)$$

Then the obtained rule has degree  $N + 2M$  [4, Theorem 5.1.3] and is defined uniquely. 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 Problem setting

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  $M$  such that a positive interpolatory quadrature rule with nodal set  $X_{N+M}$  exists.

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. It will be demonstrated that it is not always possible to add any number of nodes to an existing rule<sup>1</sup>.

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 nodes of a positive interpolatory quadrature rule, i.e. there exists a set of non-negative weights  $W_{N+1}$  such that

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

---

<sup>1</sup>Theoretically, adding  $M$  nodes with  $M \geq N + 1$  is always possible, as it is possible to add a Gaussian quadrature rule of  $N + 1$  nodes to a quadrature rule of  $N + 1$  nodes to obtain a positive interpolatory quadrature rule of  $2N + 2$  nodes. However, in this case the weights of the quadrature rule become all equal to zero, so this construction is not of much practical interest.

Here,  $w_k^{(N+1)}$  are the weights in the set  $W_{N+1}$  and  $\mu_j$  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  $\mu_{N+1}$ . 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 moreover as example a straightforward quadrature rule is constructed using a heuristic that follows naturally from the derived framework.

### 3.1 Positive weight criterion

The key notion is that if the node  $x_{N+1}$  is given, a vector  $\mathbf{c} = (c_0, \dots, c_{N+1})^T$  can be constructed such that  $w_k^{(N+1)} = w_k^{(N)} + c_k$  (for  $k = 0, \dots, N+1$ ). If this vector is such that  $c_k \geq -w_k^{(N)}$ , then  $w_k^{(N+1)} \geq 0$ , which is the primary goal. 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:

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

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

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

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  $\mu_{N+1} = \int_{\Omega} x^{N+1} \rho(x) dx$  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

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

The value of  $\varepsilon_{N+1}$  can be interpreted as the approximation error of the quadrature rule with nodes  $X_N$  and weights  $W_N$  with respect to  $\mu_{N+1}$ . 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  $\boldsymbol{\varepsilon} = (0, \dots, 0, \varepsilon_{N+1})^T$ . 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

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

where  $V_k(X_{N+1})$  is equal to  $V(X_{N+1})$  with the  $k$ th column replaced by  $\varepsilon$ , 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 the Vandermonde-determinant from (2.3), the following is obtained for  $k = 0, \dots, N+1$ :

$$\begin{aligned} c_k &= \frac{\det V_k(X_{N+1})}{\det V(X_{N+1})} = (-1)^{N+k+1} \varepsilon_{N+1} \frac{\det V(X_{N+1} \setminus \{x_k\})}{\det V(X_{N+1})} \\ &= (-1)^{N+k+1} \varepsilon_{N+1} \left( \prod_{\substack{0 \leq i < j \leq N+1 \\ i, j \neq k}} (x_j - x_i) \right) / \left( \prod_{0 \leq i < j \leq N+1} (x_j - x_i) \right) \\ &= \varepsilon_{N+1} / \left( \prod_{\substack{j=0 \\ j \neq k}}^{N+1} (x_k - x_j) \right). \end{aligned} \quad (3.3)$$

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 results into the following theorem.

**Theorem 1.** *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*

$$-\varepsilon_{N+1} / \left( \prod_{\substack{j=0 \\ j \neq k}}^{N+1} (x_k - x_j) \right) \leq w_k^{(N)}, \text{ for } k = 0, \dots, N+1. \quad (3.4)$$

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  $k = 0, \dots, N$ ).

## 3.2 Quadrature rule adjustments

Theorem 1 describes a necessary and sufficient condition for a quadrature rule to have positive weights if both  $x_{N+1}$  and  $\varepsilon_{N+1}$  are known. A main novelty of this work is to employ a geometrical interpretation of (3.4), 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 simplify (3.4) and discuss the geometrical relation between the new node  $x_{N+1}$  and the quadrature error  $\varepsilon_{N+1}$ . In Section 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 1 derived in Section 3.2.1. Based on Theorem 1, it is also possible to derive a removal adjustment, where one node is removed from the quadrature rule and a positive interpolatory quadrature rule is obtained. This has been studied extensively in [2] and is therefore not repeated here.

### 3.2.1 Geometry of nodal addition

The inequalities from (3.4) are  $N+2$  linear inequalities in  $x_{N+1}$  and  $\varepsilon_{N+1}$ . This can be seen by rewriting (3.3) as follows:

$$c_k \prod_{\substack{j=0 \\ j \neq k}}^{N+1} (x_k - x_j) = \varepsilon_{N+1}, \text{ for } k = 0, \dots, N+1. \quad (3.5)$$

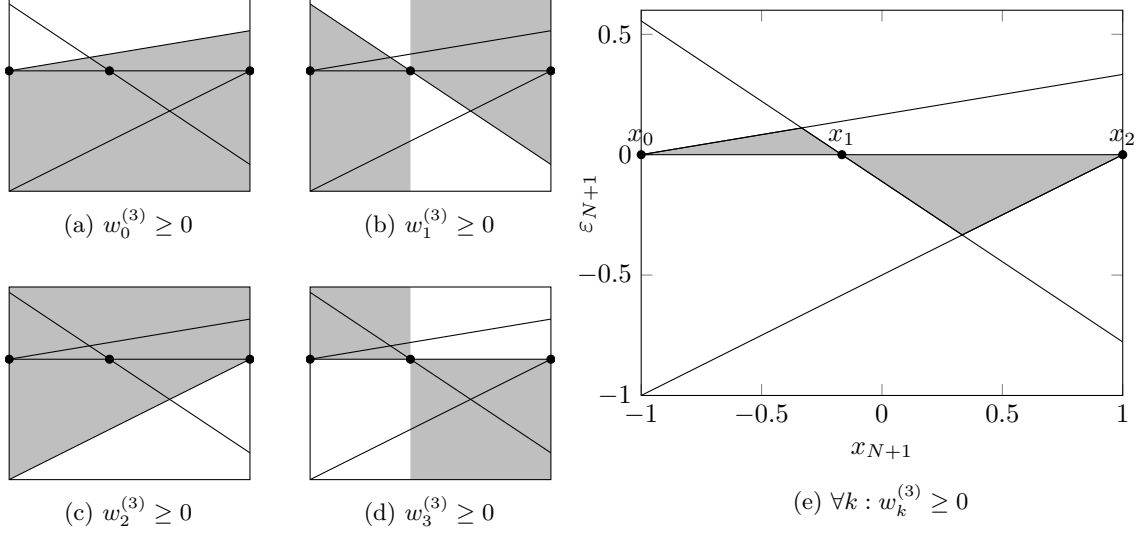


Figure 1: The quadrature rule error  $\varepsilon_{N+1}$  versus the new node  $x_{N+1}$  using the quadrature rule  $X_N = \{-1, -1/6, 1\}$  and  $\rho \equiv 1/2$ . The solid lines depict locations such that one weight becomes zero. *Left:* regions where individual weights are positive. *Right:* region where all weights are positive, which is the intersection of the left figures.

If two values of  $x_{N+1}$ ,  $c_k$  (for  $k = 0, \dots, N+1$ ), or  $\varepsilon_{N+1}$  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 use that for  $k = 0, \dots, N$  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 (3.4) and requiring  $w_k^{(N)} + c_k \geq 0$  inequalities of the following form are obtained:

$$\begin{aligned} \varepsilon_{N+1} &\leq -w_k^{(N)}(x_k - x_{N+1}) \prod_{\substack{j=0 \\ j \neq k}}^N (x_k - x_j) && \text{if } \prod_{\substack{j=0 \\ j \neq k}}^{N+1} (x_k - x_j) \leq 0, \\ \varepsilon_{N+1} &\geq -w_k^{(N)}(x_k - x_{N+1}) \prod_{\substack{j=0 \\ j \neq k}}^N (x_k - x_j) && \text{if } \prod_{\substack{j=0 \\ j \neq k}}^{N+1} (x_k - x_j) \geq 0. \end{aligned} \quad (3.6)$$

These are linear inequalities describing the relation between  $x_{N+1}$  and  $\varepsilon_{N+1}$  such that  $w_k^{(N+1)} \geq 0$  for  $k = 0, \dots, N$ . For  $k = N+1$  it holds that  $w_k^{(N)} = 0$ , so by using that  $c_k = w_k^{(N+1)}$ , (3.5) translates to:

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

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.

These inequalities are visualized as functions from  $x_{N+1}$  to  $\varepsilon_{N+1}$  in Figure 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 (3.6) or (3.7)). The region where individual weights are positive are shaded in subfigures 1a, 1b, 1c, and 1d. Subfigure 1e 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 (3.6). The characteristics of the last inequality (subfigure 1d) solely depend on the location of  $x_{N+1}$  with respect to the other nodes. A combination of all inequalities (subfigure 1e) 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  $\mu_{N+1}$  is already correctly integrated by the quadrature rule.

These figures can be interpreted in two ways. Firstly, if a new node  $x_{N+1}$  is given, an upper bound and lower bound on  $\varepsilon_{N+1}$  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  $\varepsilon_{N+1}$  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.  $\varepsilon_{N+1}$  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 node 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

Adding a node is determining an  $x_{N+1}$  that solves (3.6) and (3.7) if  $\varepsilon_{N+1}$  is known. Mathematically, this is equivalent to combining the solutions  $x_{N+1}^{[k]}$  (indexed by  $[k]$  with  $k = 0, \dots, N$ ) of the following problems:

$$\varepsilon_{N+1} = -w_k^{(N)}(x_k - x_{N+1}^{[k]}) \underbrace{\prod_{\substack{j=0 \\ j \neq k}}^N (x_k - x_j)}_{\ell'_N(x_k)}, \text{ for } k = 0, \dots, N, \quad (3.8)$$

hence if  $w_k^{(N)} \neq 0$ :

$$x_{N+1}^{[k]} = \frac{\varepsilon_{N+1} + w_k^{(N)} x_k \ell'_N(x_k)}{w_k^{(N)} \ell'_N(x_k)}, \text{ for } k = 0, \dots, N.$$

Here we used  $\ell'_N$  to make the notation more compact. The nodes  $x_{N+1}^{[k]}$  are such that, if added to the quadrature rule, a (possible negative) interpolatory quadrature rule is obtained with  $w_k^{(N+1)} = 0$ . Those  $x_{N+1}^{[k]}$  that yield positive quadrature rules describe intervals and all nodes in these intervals can be added to the rule such that positive weights are obtained. The bounds of these intervals, i.e. the solutions of (3.8), are depicted in Figure 2a as open circles. The same quadrature rule example as in the previous section is used and a constant valued  $\rho$  is considered. In this case, the three solutions 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 intervals  $(-\infty, -5/3]$  or  $[0, 7/9]$  yields a positive interpolatory quadrature rule. Restricting  $x_{N+1}$  to the set  $\Omega$  further reduces the number of possible intervals.

The node  $x_{N+1}^{[k]}$  does not exist 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 (3.8)) 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 into a singular Vandermonde-matrix (which contradicts the theory developed so far), so we do not further study this specific case.



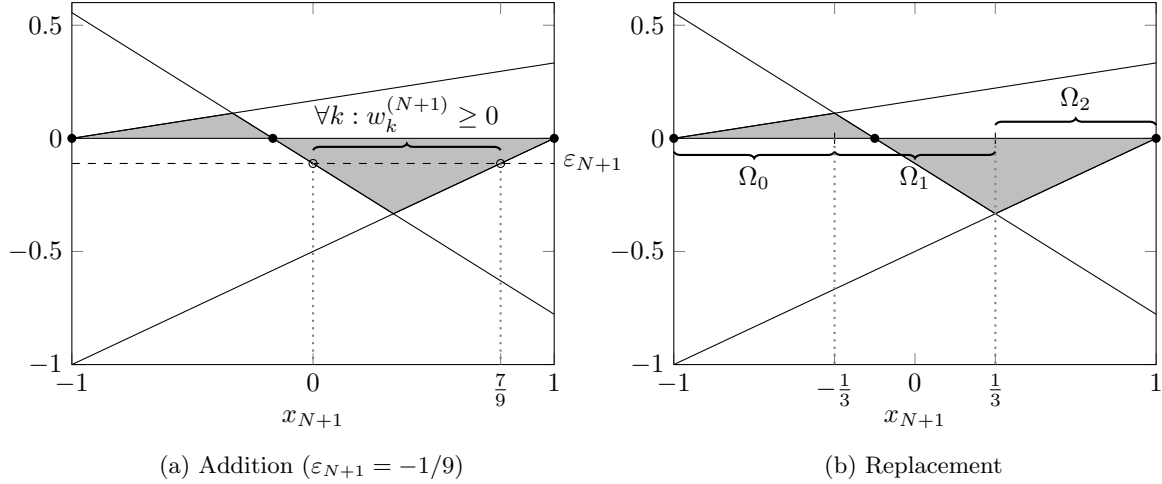


Figure 2: Addition of a new node to and replacement of an existing node within the quadrature rule  $X_N = \{-1, -1/6, 1\}$  and  $\rho \equiv 1/2$ .

If  $\Omega = \mathbb{R}$  and the number of nodes is odd, it is always possible to add a single node to a quadrature rule. Geometrically this means that the leftmost and rightmost shaded region grow to infinity and minus infinity respectively (or vice versa). However, in any other case (i.e. that of a bounded  $\Omega$  or even number of nodes) adding a single node to a quadrature rule is not always possible. An example is the following interpolatory quadrature rule:

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

Note that this example can be obtained straightforwardly by adding the node  $1/11$  to the previous stated quadrature rule 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  $\mu_{N+1}$  is not used, as the number of nodes and the degree of the rule do not change. This allows to freely choose  $\varepsilon_{N+1}$ .

The relation between  $\varepsilon_{N+1}$  and  $x_{N+1}$  is already derived, so by reconsidering (3.8) with the goal to determine both  $x_{N+1}$  and all  $\varepsilon_{N+1}^{[k]}$  (indexed by  $[k]$  with  $k = 0, \dots, N$ ) that make  $w_k^{(N+1)} = 0$  the following expressions are obtained:

$$\varepsilon_{N+1}^{[k]} = -w_k^{(N)}(x_k - x_{N+1}) \prod_{\substack{j=0 \\ j \neq k}}^N (x_k - x_j), \text{ for } k = 0, \dots, N. \quad (3.9)$$

We will interpret this expression as a function of  $x_{N+1}$ , denoted by  $\varepsilon_{N+1}^{[k]} : \Omega \rightarrow \mathbb{R}$ . These functions represent the solid lines in Figure 2b. 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.

Graphically, it can be seen 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 node  $x_k$  can be found by determining the bounds on the shaded region and observing which node belongs to the obtained bound. The sets  $\Omega_k$  describe this property mathematically: if an  $x_{N+1} \in \Omega_k$  is added to the quadrature rule, then  $X_N \cup \{x_{N+1}\} \setminus \{x_k\}$  forms the nodes of a positive interpolatory quadrature rule. The fact that  $\Omega = \bigcup_{k=0}^N \Omega_k$  follows from Figure 2b. Mathematically, these sets can be denoted (possibly less intuitively)

in the following way:

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

A special case is given by the intersection of the lines in Figure 2b. Notice that  $\Omega_k$  is a closed interval, because  $\prod_{j \neq k}^N (x_k - x_j) = 0$  forms the boundary and is included in the interval. Combining this with  $\Omega = \bigcup_{k=0}^N \Omega_k$  yields that for each  $\Omega_k$  there exists at least one other  $\Omega_j$  such that  $\Omega_k \cap \Omega_j$  is not empty. This means that for  $x_{N+1} \in \Omega_k \cap \Omega_j$ , the quadrature rule  $X_N \cup \{x_{N+1}\} \setminus \{x_k\} \setminus \{x_l\}$  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. An example of such a rule would be  $X_N = \{-1, 1/3\}$ , obtained by adding  $1/3$  to the quadrature rule example of Figure 2b. Such rules have a non-trivial high degree and are therefore more accurate than interpolatory quadrature rules without this property.

This special case can be further generalized by allowing negative weights, which will be very useful in light of multiple node addition discussed in Section 4. To introduce this formally, let the node  $x_{(k,l)}$  be such that  $X_N \cup \{x_{(k,l)}\} \setminus \{x_k\} \setminus \{x_l\}$  is a (possibly negative) interpolatory quadrature rule. This node exists if and only if it satisfies the following linear equality, that follows from (3.9):

$$-w_k^{(N)}(x_k - x_{(k,l)}) \prod_{\substack{j=0 \\ j \neq k}}^N (x_k - x_j) = -w_l^{(N)}(x_l - x_{(k,l)}) \prod_{\substack{j=0 \\ j \neq l}}^N (x_l - x_j). \quad (3.10)$$

Notice that this linear inequality does not necessarily have a solution. Graphically there is no solution if the lines from Figure 2b through  $x_k$  and  $x_l$  are parallel.

The node  $x_{(k,l)}$  is independent from  $x_k$  and  $x_l$ . This is not evident, as (3.10) depends on these nodes. However, it can be demonstrated by using that the rule interpolatory, which yields:

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

Here,  $L_k(x)$  is the  $k$ th Lagrange basis polynomial. Replacing this expression in (3.10) and using that  $\ell_N'(x_k) = \prod_{j \neq k} (x_k - x_j)$  yields an equality that can be simplified to the following:

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

This expression is in fact a Patterson extension (consider (2.8) 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 [24, 25] 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. As discussed previously, there does not always exist a single node that can be added such that positive weights are obtained, in which case a heuristic can be used to select the new node. Constructing heuristics is not the main focus of this article, but an example is discussed in Section 3.3.

#### 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  $\varepsilon_{N+1}$  and  $x_{N+1}$ . As example, 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 (3.8) (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  $X_{N-2} = X_N \setminus \{x_l\} \setminus \{x_k\}$ . 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  $X_{N-2} \cup \{x_{(k,l)}\}$ .

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 (3.10) 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 (3.10). 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 Heuristic 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, a straightforward heuristic is presented that adds a node that keeps the condition number  $\kappa_N$  minimal.

The idea is to, given a quadrature rule, consider all nodes  $x_{(k,l)}$  (for  $0 \leq k < l \leq N$ ) as defined by (3.10). The nodes  $x_{(k,l)}$  are the intersections of the solid lines of Figures 2a and 2b and they form the Patterson extension for  $M = 1$  of the quadrature rules with nodes  $X_N \setminus \{x_k, x_l\}$  (see Section 3.3.1). From all these nodes  $\{x_{(k,l)}\}$ , that node is added that minimizes the condition number of the quadrature rule:

$$x_{N+1} = \arg \min_{x_{(k,l)}} \kappa_{N+1} = \arg \min_{x_{(k,l)}} \sum_{k=0}^{N+1} |w_k^{(N+1)}|, \text{ for } 0 \leq k < l \leq N, \quad (3.11)$$

where  $w_k^{(N+1)}$  are the weights of the interpolatory quadrature rule  $X_N \cup \{x_{N+1}\}$ . This heuristic yields a quadrature rule with positive weights if that is possible. If there are multiple values of  $x_{N+1}$  such that the obtained rule has positive weights, the node is selected such that the obtained quadrature rule has strictly positive weights (i.e. no weights equal to zero), as having nodes with weight equal to zero limits possible values of  $\varepsilon_{N+2}$  (as one of the lines in Figure 2a gets slope zero).

The results of this procedure for the uniform distribution with  $\Omega = [-1, 1]$  and the Beta(10, 10) distribution with  $\Omega = [0, 1]$  are depicted in Figure 3. In both cases the algorithm is initiated with the quadrature rule example discussed before (rescaled to  $[0, 1]$  for the Beta distribution). Even though the quadrature rules do get negative weights, the absolute magnitude remains small (in all cases  $\kappa_N \leq 6$ ). When allowing addition of multiple nodes, it is possible to keep all weights positive, as will be demonstrated in Section 4. We will compare the performance of this quadrature rule with that of nested, positive, and interpolatory quadrature rules in Section 5.

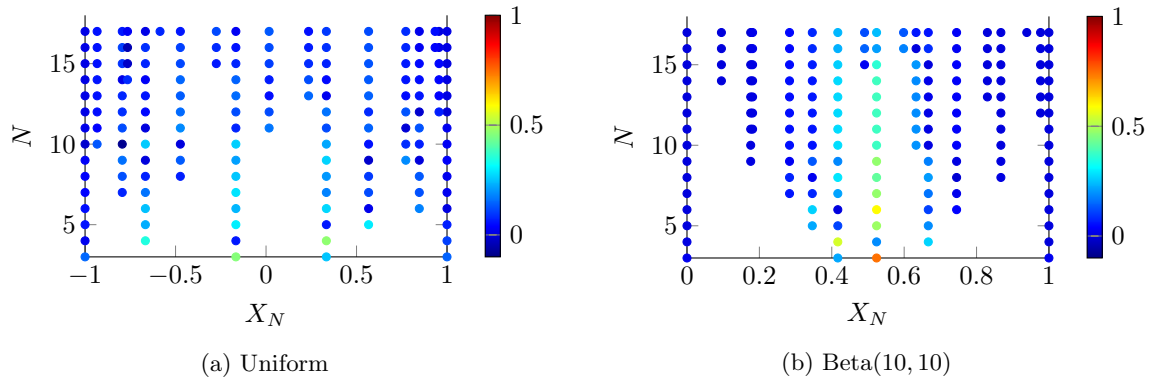


Figure 3: Nested quadrature rules constructed using the heuristic from (3.11) that minimizes the condition number.

## 4 Addition of multiple nodes

In the previous section a counterexample of a positive interpolatory quadrature rule is discussed that can not 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 2.3: 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 nodes of a positive interpolatory quadrature rule.

The first step is to extend the derivation of Section 3.1 for the addition of multiple nodes. Applying Cramer's rule is slightly more complicated in this case and requires a bit more bookkeeping, but the key principles are the same. With the theory that is derived in the 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 a positive interpolatory quadrature rule.

Let  $w_k^{(N)}$  for  $k = 0, \dots, N$  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  $\mathbf{c} = (c_0, \dots, c_N, c_{N+1}, \dots, c_{N+M})^T$  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:

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

and

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

where  $\varepsilon_j$  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  $\varepsilon = (0, \dots, 0, \varepsilon_{N+1}, \dots, \varepsilon_{N+M})^\top$ , 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  $k$ th column (indexed from 0) replaced by  $\varepsilon$ . 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  $j$ th row and  $k$ th column, 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 [9, 20], and it is well-known that the ratio of determinants obtained in this expression is an elementary symmetric polynomial. The  $k$ th elementary symmetric polynomial is generally defined as the sum of all monomial permutations of length  $k$ , that is as follows:

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

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  $c_k$ :

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

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

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

We are now in a position to formulate a theorem in similar form as Theorem 1, but then for multiple nodes.

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

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

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 1. So Theorem 2 is indeed a strict generalization of Theorem 1.

## 4.2 Quadrature rule adjustments

Theorem 2 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 (3.6), i.e. for  $k = 0, \dots, N + M$ :

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

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 Section 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 Figure 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). For example, let  $\rho \equiv 1/2$  with  $\Omega = [-1, 1]$ . In Figure 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 into one weight equal to zero. Adding the coordinates on the corners, depicted by the open circles (i.e. “the boundary of the boundary”), results into 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 into 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 these insights. 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 Figure 4). Hence if all corners of those regions are determined (depicted as open circles in Figure 4), the full region can be explored straightforwardly.

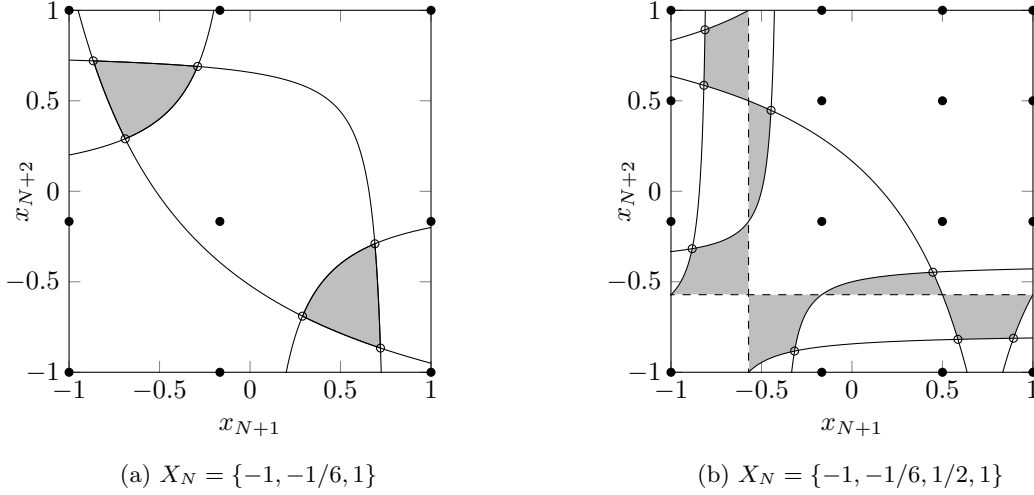


Figure 4: Two examples of addition of two nodes to a quadrature rule. Choosing the two nodes in a shaded area yields positive weights. Choosing the two nodes on the open circles yields two weights equal to zero and positive weights. Dashed lines correspond to a zero weight for  $x_{N+1}$  or  $x_{N+2}$ .

As these corners 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 Figure 4. Moreover, they describe the boundary of the set of nodes that can be added to the quadrature rule.

The desired nodes  $x_{N+1}, \dots, x_{N+M}$  can be determined by calculating the Patterson extension of the interpolatory quadrature rule with the nodes  $X_N \setminus \{x_{k_1}, \dots, x_{k_M}\}$ , for which efficient techniques exist [17, 18, 25]. However, such techniques require that  $M$  must be known a priori and they do not provide a simple geometrical interpretation. Therefore, a new algorithm that can be used to determine minimal  $M$  is presented here, which is based on the geometrical framework derived so far.

We start by solving a slightly easier problem. Assume  $\varepsilon_{N+1} = \dots = \varepsilon_{N+M-1} = 0$  and  $\varepsilon_{N+M} \neq 0$ . Notice that, if  $\varepsilon_{N+M}$  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 Figure 4) is obtained. This can be exploited to determine the desired nodes, as only the value of  $\varepsilon_{N+M}$  imposes a condition on the nodes  $x_{N+1}, \dots, x_{N+M}$ .

The nodes that yield  $w_{k_1}^{(N+M)} = \dots = w_{k_M}^{(N+M)} = 0$  can be found by applying Theorem 2 with  $c_{k_i} = -w_{k_i}^{(N)}$  for all  $i$  or by consecutively applying Theorem 1. In both cases, the following is obtained:

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

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  $x_{N+1}, \dots, x_{N+M}$ :

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

---

**Algorithm 1** Determining  $X_{N+M}$  with zero weights

---

**Input:** Interpolatory quadrature rule  $X_N$ ,  $W_N$ , indices  $k_1, \dots, k_M$ .

**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**  $k = k_1, \dots, k_M$  **do**
- 3:     Determine  $\hat{\ell}_m$  such that  $\hat{\ell}_m(x) = x^m + q_m(x)$  (see text) and

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

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

which translates the system above to

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

If the nodal polynomial  $\hat{\ell}_M$  is known, its roots equal  $x_{N+1}, \dots, x_{N+M}$ . 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:

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

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 [1]). If  $q_M$  is determined, the roots of the polynomial  $\hat{\ell}_M(x) = q_M(x) + x^M$  are the nodes  $x_{N+1}, \dots, x_{N+M}$ . By construction these nodes are such that  $w_{k_i}^{(N+M)} = 0$  for  $i = 1, \dots, M$ .

Even though assuming  $\varepsilon_{N+1} = \dots = \varepsilon_{N+M-1} = 0$  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 a replacement from 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 Figure 4. It is an iterative algorithm: a replacement of  $M$  nodes is determined using a replacement of  $M-1$  nodes. Geometrically, the algorithm iterates over the dimension of the figure and iteratively determines a set of nodes that can be used as a replacement.

The full algorithm is outlined in Algorithm 1. By iterating over all possible sorted sequences  $(k_1, \dots, k_M)$ , 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 1. 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  $\varepsilon_{N+m} = 0$ .

### 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  $X_{N+M}$  and secondly to explore *all* such  $M$  nodes (i.e. the shaded areas of Figure 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  $k_1, \dots, k_M$  with  $k_1 < \dots < k_M$ . 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 2.3). Algorithm 1 can be used to efficiently determine  $M$ , as results from previous iterations can be reused. To see this, suppose a quadrature rule is given and by applying Algorithm 1 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 1 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

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

which grows fast for large  $N$ . A more efficient algorithm to determine these nodes could possibly speed up the algorithm, but this is outside the scope of this work.

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.

Notice that the possibility of adding  $M$  nodes to the quadrature rule does not guarantee the possibility of adding  $M + 1$  nodes to the quadrature rule. This can be observed by revisiting the quadrature rule example from the previous section, 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 Figure 5a regions are depicted where a single node can be added (similar to Figure 2a) and regions where, upon adding a node from that region, another node can be added (this is the projection of Figure 4a). The addition of the rightmost node with the latter property is depicted in Figure 5b, 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 1 are a Patterson extension of a quadrature rule with a smaller number of nodes. As the Gaussian quadrature rule is a

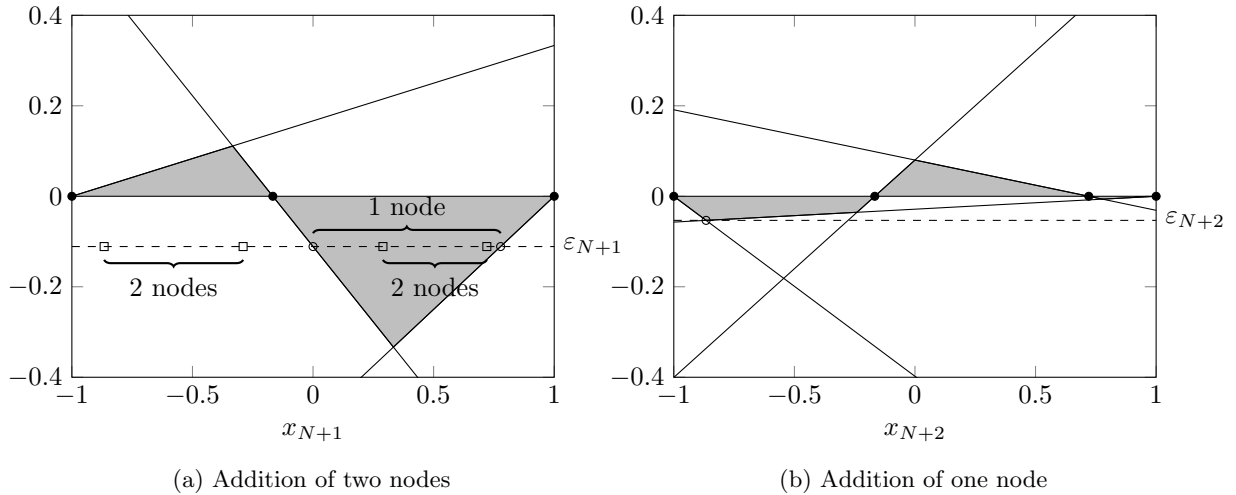


Figure 5: The addition of 2 nodes to the interpolatory quadrature rule with the nodes  $X_N = \{-1, -1/6, 1\}$ . The right quadrature rule is obtained by adding the rightmost highlighted node of the left figure (i.e. “the rightmost square”).

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 1, 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 Figure 2a and Figure 4a). 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  $x_{N+1}, \dots, x_{N+M}$  be such that the following nodes form a quadrature rule of degree  $N + M$ :

$$X_N \cup \{x_{N+1}, \dots, x_{N+M}\} \setminus \{x_{k_1}, \dots, x_{k_M}\}. \quad (4.4)$$

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

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

Upon adding  $\{x_{N+1}, \dots, x_{N+M}\}$  to  $X_{N-M}$ , the nodes from (4.4) 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 special case from Algorithm 1. 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  $k_1, \dots, k_M$ , defined as follows up to a permutation:

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

By applying Algorithm 1, the nodes from (4.4) are obtained with  $M = N + 1$ , which are:

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

Hence the  $N + 1$  nodes  $x_{N+1}, \dots, x_{2N+1}$  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.

To demonstrate where these rules 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 1 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):

$$X_{N+2} = \left\{-1, -\frac{1}{6}, 1, \frac{1}{5}(-1 - \sqrt{6}), \frac{1}{5}(-1 + \sqrt{6})\right\},$$

$$W_{N+2} = \left\{0, 0, \frac{1}{9}, \frac{1}{36}(16 + \sqrt{6}), \frac{1}{36}(16 - \sqrt{6})\right\}.$$

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 into 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 [12, 18], 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 1 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 2.3 and examples of such quadrature rules are depicted in Figure 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 a heuristic was used: the rules obtained in this section are positive, 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 the Section 4.2.3, as shown in Figure 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 leads to different behavior of  $M$ . We will further study the performance of this quadrature rule in Section 5.

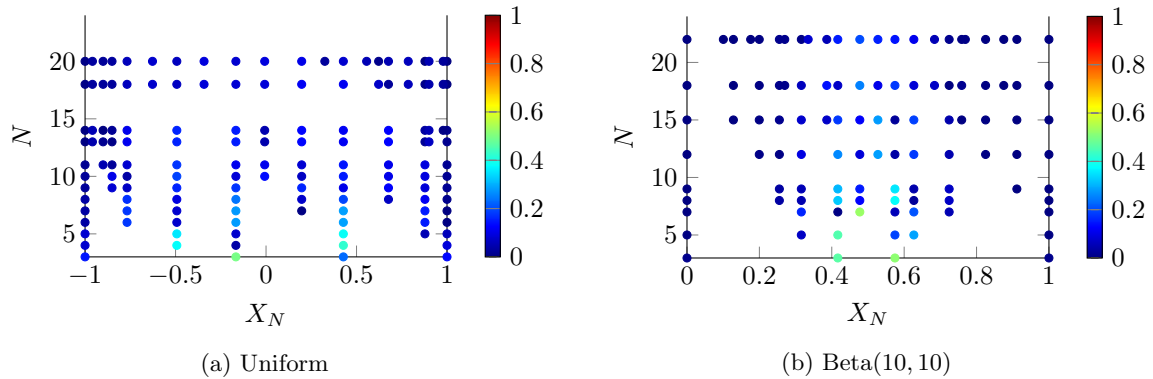


Figure 6: Nested, positive, and interpolatory quadrature rules, initialized with  $X_N = \{-1, -1/6, 1\}$  (left) or  $X_N = \{0, 5/12, 1\}$  (right).

## 5 Numerical example

This article is concerned with the construction of quadrature rules with *positive* weights and two new quadrature rules have been introduced: one based on a heuristic (with possibly negative weights) and one by randomly adding nodes ensuring positive weights. We briefly assess the numerical performance of these quadrature rules by means of Genz test functions (see Table 1). The Genz test functions [10] 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  $b$ . In this work the univariate Genz functions are considered. We restrict ourselves to the uniform distribution, as in this case the exact value of the integral of the Genz functions is known analytically.

We consider the performance of four quadrature rules:

1. The quadrature rule that is determined iteratively using (3.11), i.e. by heuristically minimizing  $\kappa_N$  (see Figure 3a). This rule was discussed in Section 3.3.2 and is a nested interpolatory quadrature rule. The rule is initialized with the quadrature rule nodes  $X_N = 1/2 \widehat{X}_N + 1/2$  with  $\widehat{X}_N = \{-1, -1/6, 1\}$  (i.e. the example nodes discussed before are 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 Figure 6a). This rule was discussed in Section 4.3.2 and is a nested, positive, and interpolatory quadrature rule. The rule is initialized in the same way as the heuristic quadrature rule (see previous point).
3. The Clenshaw–Curtis quadrature rule [5], where the nodes  $X_N$  are defined explicitly using (2.7). It is well-known that these nodes have positive weights if the distribution under consideration is uniform. This quadrature rule is nested for specific levels, positive, and interpolatory.
4. The Gaussian quadrature rule [12], 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 into 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  $u = u_k$  with  $k = 1, \dots, 6$ , so  $u$  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.

It is instructive to compare the error with the upper bound from (2.6) for positive weights:

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

where we use that  $\mu_0 = 1$  in our test cases. This error is determined using the algorithm of Remez [30, Chap. 3], with the implementation from `chebfun` [8]. Convergence results for the uniform distribution

Table 1: The test functions from Genz [10], which depend on the shape and translation parameter  $a$  and  $b$ .

Integrand Family	Attribute
$u_1(x) = \cos(2\pi b + ax)$	Oscillatory
$u_2(x) = (a^{-2} + (x - b)^2)^{-1}$	Product Peak
$u_3(x) = (1 + ax)^{-2}$	Corner Peak
$u_4(x) = \exp(-a^2(x - b)^2)$	Gaussian
$u_5(x) = \exp(-a x - b )$	$C_0$ function
$u_6(x) = \begin{cases} 0 & \text{if } x > b \\ \exp(ax) & \text{otherwise} \end{cases}$	Discontinuous

$\rho \equiv 1$  in  $\Omega = [0, 1]$  are gathered in Figure 7. The first iteration starts at 4, because the two algorithms proposed in this article are initialized with a quadrature rule of three nodes.

Notice that in all cases each quadrature rule remains 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  $N$  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  $E_N \sim 1/N$ ). Notice that the negativity of the weights of the heuristic addition algorithm becomes apparent in this case, as the error significantly grows at approximately 10 and 25 nodes. As the convergence results are averaged, this growth is not due to a specific unfortunate choice of  $a$  and  $b$ . 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  $\infty$ -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).

## 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

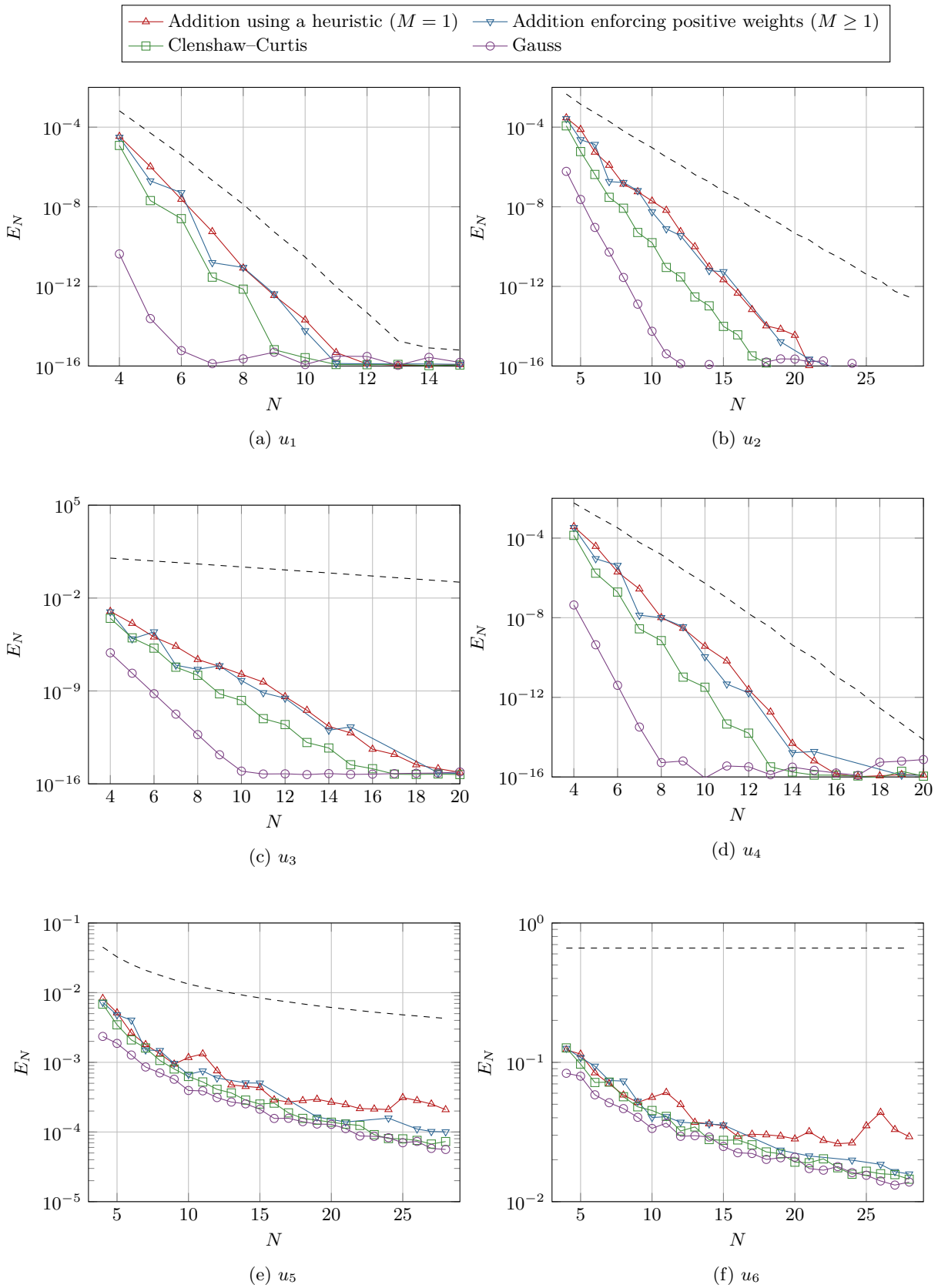


Figure 7: Convergence of the Genz test functions using various quadrature rule techniques. The absolute error of the best approximation polynomial (i.e.  $\inf_{\varphi \in \mathbb{P}(N)} \|u - \varphi\|_\infty$ ) is dashed.

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 quadrature rules and two examples have been discussed: one based on consecutively adding one node and one based on consecutively adding multiple nodes.

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 [28], 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.

## References

- [1] J.-P. Berrut and L. N. Trefethen. Barycentric Lagrange interpolation. *SIAM Review*, 46(3):501–517, 2004. doi:10.1137/s0036144502417715.
- [2] L. M. M. van den Bos, B. Koren, and R. P. Dwight. Non-intrusive uncertainty quantification using reduced cubature rules. *Journal of Computational Physics*, 332:418–445, 2017. doi:10.1016/j.jcp.2016.12.011.
- [3] R. Bourquin. Exhaustive search for higher-order Kronrod–Patterson extensions. Technical Report 2015-11, ETH Zürich, 2015.
- [4] H. Brass and K. Petras. *Quadrature Theory*, volume 178 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2011. doi:10.1090/surv/178.
- [5] C. W. Clenshaw and A. R. Curtis. A method for numerical integration on an automatic computer. *Numerische Mathematik*, 2(1):197–205, 1960. doi:10.1007/bf01386223.
- [6] R. Cools. Constructing cubature formulae: the science behind the art. *Acta Numerica*, 6:1, 1997. ISSN 1474-0508. doi:10.1017/s0962492900002701.
- [7] G. Dahlquist and Å. Björck. *Numerical Methods in Scientific Computing: Volume 1*. Society for Industrial and Applied Mathematics, 2008. ISBN 978-0-898716-44-3.
- [8] T. A. Driscoll, N. Hale, and L. N. Trefethen. *Chebfun Guide*. Pafnuty Publications, 2014.
- [9] W. Gautschi. On inverses of Vandermonde and confluent Vandermonde matrices. *Numerische Mathematik*, 4:117–123, 1962. doi:10.1007/bf01386302.
- [10] A. Genz. Testing multidimensional integration routines. In *Proceedings of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation*, pages 81–94. Elsevier North-Holland, 1984.
- [11] A. Genz and B. D. Keister. Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight. *Journal of Computational and Applied Mathematics*, 71(2):299–309, 1996. ISSN 0377-0427. doi:10.1016/0377-0427(95)00232-4.
- [12] G. H. Golub and J. H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of Computation*, 23(106):221–230, 1969. doi:10.1090/s0025-5718-69-99647-1.
- [13] T. Hasegawa, H. Sugiura, and T. Torii. Positivity of the weights of extended Clenshaw–Curtis quadrature rules. *Mathematics of Computation*, 60(202):719–719, 1993. doi:10.1090/s0025-5718-1993-1176710-2.

- [14] B. A. Ibrahimoglu. Lebesgue functions and Lebesgue constants in polynomial interpolation. *Journal of Inequalities and Applications*, 2016(1):93, 2016. doi:10.1186/s13660-016-1030-3.
- [15] D. K. Kahaner and G. Monegato. Nonexistence of extended Gauss–Laguerre and Gauss–Hermite quadrature rules with positive weights. *Zeitschrift für angewandte Mathematik und Physik*, 29(6): 983–986, 1978. ISSN 1420-9039. doi:10.1007/bf01590820.
- [16] D. K. Kahaner, J. Waldvogel, and L. W. Fullerton. Addition of points to Gauss–Laguerre quadrature formulas. *SIAM Journal on Scientific and Statistical Computing*, 5(1):42–55, 1984. ISSN 2168-3417. doi:10.1137/0905003.
- [17] D. P. Laurie. Anti-Gaussian quadrature formulas. *Mathematics of Computation*, 65(214):739–748, 1996. ISSN 0025-5718. doi:10.1090/s0025-5718-96-00713-2.
- [18] D. P. Laurie. Calculation of Gauss–Kronrod quadrature rules. *Mathematics of Computation*, 66(219):1133–1146, 1997. doi:10.1090/s0025-5718-97-00861-2.
- [19] J. Ma, V. Rokhlin, and S. Wandzura. Generalized Gaussian quadrature rules for systems of arbitrary functions. *SIAM Journal on Numerical Analysis*, 33(3):971–996, 1996. ISSN 1095-7170. doi:10.1137/0733048.
- [20] N. Macon and A. Spitzbart. Inverses of Vandermonde matrices. *The American Mathematical Monthly*, 65(2):95–100, 1958. doi:10.2307/2308881.
- [21] G. Monegato. A note on extended Gaussian quadrature rules. *Mathematics of Computation*, 30(136): 812–812, 1976. doi:10.1090/s0025-5718-1976-0440878-3.
- [22] G. Monegato. Positivity of the weights of extended Gauss–Legendre quadrature rules. *Mathematics of Computation*, 32(141):243–243, 1978. ISSN 0025-5718. doi:10.1090/s0025-5718-1978-0458809-0.
- [23] G. Monegato. An overview of the computational aspects of Kronrod quadrature rules. *Numerical Algorithms*, 26(2):173–196, 2001. ISSN 1017-1398. doi:10.1023/a:1016640617732.
- [24] T. N. L. Patterson. The optimum addition of points to quadrature formulae. *Mathematics of Computation*, 22(104):847–847, 1968. doi:10.1090/s0025-5718-68-99866-9.
- [25] 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 Transactions on Mathematical Software*, 15(2):123–136, 1989. ISSN 0098-3500. doi:10.1145/63522.63523.
- [26] F. Peherstorfer. Characterization of positive quadrature formulas. *SIAM Journal on Mathematical Analysis*, 12(6):935–942, 1981. doi:10.1137/0512079.
- [27] G. Pólya. Über die Konvergenz von Quadraturverfahren. *Mathematische Zeitschrift*, 37(1):264–286, 1933. doi:10.1007/bf01474574.
- [28] J. A. Shohat and J. D. Tamarkin. *The problem of moments*. Number 1 in Mathematical Surveys. American Mathematical Society, 1943.
- [29] L. N. Trefethen. *Spectral Methods in MATLAB*. Society for Industrial & Applied Mathematics (SIAM), 2000. doi:10.1137/1.9780898719598.
- [30] G. A. Watson. *Approximation Theory and Numerical Methods*. John Wiley & Sons Ltd, 1980. ISBN 0471277061.