

Comparing algebraic cubature rules on spline curved elements

Alvise Sommariva^a, Marco Vianello^a

^a*Dept. of Mathematics, University of Padova, via Trieste 63, 35121 Padova (Italy)*

Abstract

We compare four methods for the construction of algebraic cubature rules on planar elements, whose boundary is tracked by splines. The methods, that we have developed over the last two decades, are based on Green theorem together with some cornerstones of polynomial approximation theory: Gaussian quadrature, Tchakaloff theorem, discretized Chebyshev expansion (hyperinterpolation), Fekete-like interpolation. We discuss their advantages and drawbacks in view of the application to curved polytopal element methods. We have also made freely available at a single site the corresponding open-source Matlab codes.

Keywords: Algebraic cubature rules, curved polytopal element methods, spline curved elements.

1. Introduction

The availability of efficient cubature methods is a relevant issue within FEM and VEM methods on nonstandard elements, especially for curved polytopal elements in high-order instances. Indeed, computation of stiffness and mass matrices can still represent a bottleneck, that has to be coped with by specialized cubature rules. During the last two decades, a devoted literature has been growing on this topic, with some attention on avoiding subtriangulations or in general subtessellations; with no pretence of exhaustivity, we may quote [1, 3, 4, 9, 10, 11, 15, 16, 17, 22, 24, 27, 30, 36, 39] and the references therein.

In this respect, the main purpose of this paper is to give a guided tour in the use of some specific codes, that we developed over the last twenty years, for algebraic cubature on a general *curved element* $E \subset \mathbb{R}^2$ (either convex or concave), in the case when its *piecewise regular Jordan boundary* ∂E is tracked by splines. Such codes produce weights $\{w_s\}$ and nodes $\{P_s = (x_s, y_s)\}$ such that

$$\int_E f(x, y) dx dy = \sum_{s=1}^M w_s f(P_s) \quad \forall f \in \mathbb{P}_n, \quad (1)$$

(where $\mathbb{P}_n$ denotes the space of bivariate polynomials with total degree not exceeding n), and are here named correspondingly to the construction approach:

- **Spline-Gauss** (2007-2009) [35, 36]: collection of product-like Gaussian rules by Green theorem and numerical primitives (*possible negative weights and external nodes on concave elements, high cardinality, empirical stability in general and theoretical stability on convex elements, low construction cost*).
- **Spline-Tchakaloff** (2021 & 2024) [39, 40]: moment-matching by Green theorem and NonNegative Least Squares, supported at Tchakaloff sets (*subtessellation-free, positive weights and internal nodes, low cardinality, theoretical stability, high construction cost*).

*Corresponding author: Marco Vianello

Email addresses: alvise@math.unipd.it (Alvise Sommariva), marcov@math.unipd.it (Marco Vianello)

- **Spline-Chebyshev** (2025-2026) [27, 28, 41]: cheap moment-matching by Green theorem and discretized Chebyshev orthogonal expansion (hyperinterpolation), supported at a minimal cubature set for the Chebyshev measure in a bounding box (*subtessellation-free, some external nodes and possible negative weights, low cardinality, theoretical stability, low construction cost*).
- **Spline-Fekete** (2009 & 2026) [37]: moment-matching by Green theorem and QR factorization with column pivoting, supported at approximate Fekete sets (*subtessellation-free, possible negative weights but internal nodes, low cardinality, empirical stability, moderate construction cost*).

By stability of the cubature rule we mean the fact that $\|\mathbf{w}\|_1$ is bounded with n , and more precisely that, also in the presence of some negative weights, the stability parameter

$$\sigma = \frac{\sum_s |w_s|}{\sum_s w_s} = \frac{\sum_s |w_s|}{\text{area}(E)} \geq 1 \quad (2)$$

stays relatively close to the optimal value 1 (positive weights), at least for the degrees of interest in FEM and VEM applications.

In the sequel we give a brief description of the underlying method for each code, pointing out its theoretical foundations, as well as advantages and drawbacks. Moreover, we illustrate the performance on some test elements. The corresponding open-source codes are implemented in Matlab and freely available at [34].

2. The four cubature rules

The starting point of all the methods is that the boundary ∂E of the element is piecewise regular, and each regular arc is parametrized by a number of sampling points with a spline curve of local degree d_i . Namely

$$\begin{aligned} \partial E &= \bigcup_{i=1}^m \Gamma_i, \quad \Gamma_i = \bigcup_{j=1}^{s_i} \Gamma_{i,j}, \quad \Gamma_{i,j} = Q_{i,j-1} \frown Q_{i,j}, \\ \{Q_{i,j}\} &\subset \Gamma_i = S_i([0, 1]), \quad S_i(t) = (x_i(t), y_i(t)), \quad t \in [0, 1], \\ Q_{i,j} &= S_i(t_{i,j}), \quad 0 = t_{i,0} < t_{i,1} < \dots < t_{i,s_i} = 1, \end{aligned} \quad (3)$$

where $S_i = S_{i,d_i} \in \mathbb{P}_{d_i}$ on each parameter subinterval $[t_{i,j-1}, t_{i,j}]$ (for simplicity we consider a fixed parametrization interval $[0, 1]$ and a fixed spline degree d). The joining points $\{Q_{i,s_i}\}$ of two consecutive regular arcs are the “vertices” of such a spline curved polygon. In practice, the code user can simply give the (counterclockwise) ordered point subsequences $\{Q_{i,j}\}$ for each i together with the required spline degree d , and the spline boundary is automatically set up.

All the methods use in a substantial way Green theorem (cf. e.g. [2]) to compute integrals of polynomials, observing that a bivariate polynomial $f(x, y)$ of degree n restricted to a spline boundary, is a univariate polynomial on each polynomial spline arc $\Gamma_{i,j}$, namely the polynomial $f(x_i(t), y_i(t))$ of degree dn in the t variable. Now, if for example an x -primitive

$$\mathcal{F}(x, y) = \int f(x, y) dx \quad (4)$$

of a polynomial $f(x, y) \in \mathbb{P}_n$ is known then

$$\begin{aligned} \int_E f(x, y) dx dy &= \oint_{\partial E} \mathcal{F}(x, y) dy = \sum_{i=1}^m \int_{\Gamma_i} \mathcal{F}(x, y) dy \\ &= \sum_{i=1}^m \sum_{j=1}^{s_i} \int_{\Gamma_{i,j}} \mathcal{F}(x, y) dy = \sum_{i=1}^m \sum_{j=1}^{s_i} \int_{t_{i,j-1}}^{t_{i,j}} \mathcal{F}(S_i(t)) y'_i(t) dt \end{aligned}$$

$$= \sum_{i=1}^m \sum_{j=1}^{s_i} \sum_{h=1}^{\lfloor \frac{d_i n + 2d_i - 1}{2} \rfloor} \lambda_{i,j,h} \mathcal{F}(S_i(\tau_{i,j,h})) y'_i(\tau_{i,j,h}), \quad (5)$$

where $\{\tau_{i,j,h}\}_h$ and $\{\lambda_{i,j,h}\}_h$ are the nodes and positive weights of the Gauss-Legendre formula with exactness degree $d_i n + 2d_i - 1$ on the interval $[t_{i,j-1}, t_{i,j}]$.

2.1. Spline-Gauss

The original algorithm [36], which stemmed from a previous work on cubature for linear polygons [35], has been widely used in the finite and virtual elements literature over the last decade; cf. e.g. [11, 16], only to quote two very recent ones in the VEM framework. Though the method doesn't apparently make a preliminary subtessellation of the element, it is ultimately based on a sort of implicit covering by disjoint curved trapezoids and piecewise product Gauss-Legendre quadrature.

It works substantially as follows:

- a vertical reference line $x = a$ is fixed (in practice, any line up to a rotation)
- a primitive $\mathcal{F}$ is computed for every i, j, h at the boundary points $S_i(\tau_{i,j,h}) = (x_{i,j,h}, y_{i,j,h}) = (x_i(\tau_{i,j,h}), y_i(\tau_{i,j,h}))$ in (5) as

$$\mathcal{F}(S_i(\tau_{i,j,h})) = \int_a^{x_{i,j,h}} f(x, y_{i,j,h}) dx = \sum_{k=1}^{\lfloor n/2 \rfloor} \nu_{i,j,h,k} f(P_{i,j,h,k}), \quad \forall f \in \mathbb{P}_n,$$

with $P_{i,j,h,k} = (\xi_{i,j,h,k}, y_{i,j,h})$, where $\{\xi_{i,j,h,k}\}_k$ and $\{\nu_{i,j,h,k}\}_k$ are the nodes and weights of the Gauss-Legendre rule of exactness degree dn on the interval $[a, x_{i,j,h}]$.

- the final algebraic cubature rule on the element is

$$\begin{aligned} \int_E f(x, y) dx dy &= \sum_{i=1}^m \sum_{j=1}^{s_i} \sum_{h=1}^{\lfloor \frac{d_i n + 2d_i - 1}{2} \rfloor} \sum_{k=1}^{\lfloor n/2 \rfloor} w_{i,j,h,k} f(P_{i,j,h,k}), \quad \forall f \in \mathbb{P}_n, \\ w_{i,j,h,k} &= \lambda_{i,j,h} y'_i(\tau_{i,j,h}) \nu_{i,j,h,k}. \end{aligned} \quad (6)$$

Though the construction of the Spline-Gauss cubature rule (6) is easy and cheap, being based on univariate Gauss-Legendre quadrature, its *cardinality can be large*, namely

$$M = \text{card}(\{P_{i,j,h,k}\}) \approx \frac{n^2}{4} \sum_{i=1}^m d_i s_i, \quad (7)$$

so that M can already reach a size of the thousands for example with cubic splines ($d_i = 3$) and relatively small values of n (exactness degree) and m (number of regular sides of the curved element).

In addition, the Spline-Gauss rule has in general *possible external nodes* and some *possible negative weights*. This happens typically with concave elements or with nonoptimal choices of the reference line. Nevertheless, as proved in [36], the stability parameter σ in (2) is bounded. Moreover in practice, at least for moderate exactness degrees (up to around $n = 20$) the negative weights turns out to be few and of small size, so that the stability parameter stays relatively close to the optimal value 1. On the other hand, again in [36] it is proved that with *convex* spline boundaries the nodes result all *internal* and the weights all *nonnegative*, simply by choosing as reference line a diameter of the element.

It is worth recalling that, when the weights of Spline-Gauss are positive, one could compress the corresponding discrete measure obtaining a Tchakaloff-like rule [42], by the NonNegative Least-Squares method proposed in [38], and then adopted for example in [11, 16] in the VEM framework. However, we prefer to describe in the next subsection a method, named Spline-Tchakaloff, that produces a Tchakaloff-like rule with guaranteed internal nodes and positive weights, irrespectively of the element shape.

2.2. Spline-Tchakaloff

The method relies on *measure compression via linear or quadratic programming*, a concept emerged in the numerical literature around the mid 2010s, independently in different deterministic and probabilistic contexts; cf. e.g. [19, 21, 29, 38, 43] with the references therein. The theoretical background is the celebrated Tchakaloff theorem [42], a cornerstone of cubature theory, which substantially asserts that

- *there exists an exact cubature formula with at most $N = \dim(\mathbb{P}_n)$ nodes and positive weights, for the integration of polynomials in $\mathbb{P}_n$ with respect to a positive Borel measure with compact support.*

An often overlooked refinement of Tchakaloff theorem is a result due to Davis and Wilhelmsen [45], ensuring that Tchakaloff-like rules on a compact domain E can be supported at a suitable subset of a so-called Tchakaloff set (a sufficiently long finite sequence of an everywhere dense sequence in E).

We can then implement the Tchakaloff-Davis-Wilhelmsen result on a spline curved element E by moment-matching in the following way:

- take the bivariate Chebyshev polynomial basis of a cartesian bounding box $B \supseteq E$ with center (c_1, c_2) and side half-lengths ℓ_1 and ℓ_2

$$f_\alpha(x, y) = T_{\alpha_1} \left(\frac{x - c_1}{\ell_1} \right) T_{\alpha_2} \left(\frac{y - c_2}{\ell_2} \right), \quad (x, y) \in B, \quad \alpha = (\alpha_1, \alpha_2), \quad 0 \leq \alpha_1 + \alpha_2 \leq n, \quad (8)$$

where $T_s(t) = \cos(s \arccos(t))$ are the Chebyshev polynomials of the first kind in $[-1, 1]$ and $(x, y) \mapsto (\ell_1 x + c_1, \ell_2 y + c_2)$ is the standard affine transformation of the bounding box into the reference square $[-1, 1]^2$; the $N = \dim(\mathbb{P}_n) = (n+1)(n+2)/2$ couples (α_1, α_2) are suitably (e.g. lexicographically) ordered

- consider the x -primitives $\mathcal{F}_\alpha(x, y) = \int f_\alpha(x, y) dx$

$$\mathcal{F}_\alpha(x, y) = \ell_1 \mathcal{F}_{\alpha_1} \left(\frac{x - c_1}{\ell_1} \right) T_{\alpha_2} \left(\frac{y - c_2}{\ell_2} \right), \quad (9)$$

where (cf. [20, §2.4.4])

$$\mathcal{F}_{\alpha_1}(t) = \frac{1}{2} \left(\frac{T_{\alpha_1+1}(t)}{\alpha_1+1} - \frac{T_{|\alpha_1-1|}(t)}{\alpha_1-1} \right), \quad \alpha_1 \neq 1, \quad \mathcal{F}_1(t) = \frac{T_2(t)}{4}$$

- compute the $N = (n+1)(n+2)/2$ Chebyshev moments as in (5)

$$\begin{aligned} m_\alpha &= \int_E f_\alpha(x, y) dx dy = \oint_{\partial E} \mathcal{F}_\alpha(x, y) dy \\ &= \sum_{i=1}^m \sum_{j=1}^{s_i} \sum_{h=1}^{\lceil dn/2 \rceil} \lambda_{i,j,h} \mathcal{F}_\alpha(S_i(\tau_{i,j,h})) y'_i(\tau_{i,j,h}), \quad 0 \leq \alpha_1 + \alpha_2 \leq n \end{aligned} \quad (10)$$

- take a subsequence $\mathcal{H}_{N_k}$ of the Halton sequence of the bounding box $B \supseteq E$, e.g. $N_k = (1 + \theta)N_{k-1}$, $k \geq 1$, for a suitable $\theta \in (0, 1]$, and compute the sequence

$$H_k = \mathcal{H}_{N_k} \cap E = \{Z_1, \dots, Z_{M_k}\} \quad (11)$$

by the *in-domain algorithm* based on the spline boundary *crossing number* as implemented in [39]; N_0 has to be chosen sufficiently large to ensure that $M_0 \approx N_0 \times \text{area}(E)/\text{area}(B) \geq N$, say $N_0 > (1 + \theta)N \times \text{area}(B)/\text{area}(E)$, where the element area can be computed by (5) with $\mathcal{F}(x, y) = x$

- compute the transposed Chebyshev-Vandermonde matrices $V_k^t = (f_\alpha(Z_\ell))_{\alpha,\ell} \in \mathbb{R}^{N \times M_k}$ and solve the NonNegative Least Squares problem (all vectors here are column vectors)

$$\|V_k^t u_k - \{m_\alpha\}\|_2^2 = \min_{0 \leq u \in \mathbb{R}^{M_k}} \|V_k^t u - \{m_\alpha\}\|_2^2, \quad k = 0, 1, 2, \dots \quad (12)$$

by the Lawson-Hanson active set method [18], until the residual goes below a given tolerance for a suitable k^* (see [38] for a discussion on the effect of a nonzero residual on the cubature error)

- select the $M = N$ positive weights $\{w_s\} = \{u_{k^*}(\ell) > 0\}$ and nodes $\{P_s\} = \{Z_\ell : u_{k^*}(\ell) > 0\}$ of the final cubature formula.

This algorithm has some appealing features: *no subtessellation* of the element is needed, the final weights are *positive*, the nodal set is *inside the element* and has *low cardinality*. The main drawback is an *high computational cost*. The bulk is given by Lawson-Hanson algorithm applied at each step k , that by construction is able to select a sparse solution with $M = N$ nonzero weights, but is based on a sequence of QR factorizations with column pivoting [7] (in practice, a sequence of sequences of QR factorizations is performed). Even when an accelerated version of Lawson-Hanson is used, adopting the concept of *deviation maximization* instead of column pivoting [12, 13], the method appears costly for an elementwise application within FEM or VEM on a single problem. The situation does not improve if moment-matching is implemented via linear programming instead of quadratic programming, as tested in [25]. On the other hand, the method becomes interesting when a given curved discretization mesh for a PDE has to be used several times, for example with *parametric or time-dependent problems*, since the cubature rule can be computed only once for each spline curved element.

2.3. Spline-Chebyshev

The Spline-Chebyshev approach aims at reducing the computational cost of Spline-Tchakaloff, still keeping a low cardinality of the cubature rule. The method was inspired by the “frugal” approach to algebraic cubature on polytopal elements, recently proposed in [17] to accelerate the computation of stiffness and mass matrices. It is still a moment-matching and subtessellation-free method, where in order to get at the same time a low computational cost and a low cardinality of the rule, one accepts a priori to support the rule also at external points, namely at suitable points of a bounding box for the element, and to have some possible negative weights.

In [17] this was done by polynomial interpolation at approximate Fekete points [37] of the bounding box, that can be computed by a single QR factorization with column pivoting of a transposed Vandermonde matrix at a suitable grid of the box. The resulting weights are not all positive, in general, but empirical stability was observed. While the method had some manifest advantages (no triangulation of the polytopal element needed, QR factorization made only once on a reference box), it also presented the drawback of solving a linear system for each element, possibly ill-conditioned due to the adoption of the monomial basis to simplify moment computation by the recursive algorithm in [1].

In [27] we proposed a similar approach, based on *Chebyshev-like hyperinterpolation* instead of interpolation. We recall that Sloan hyperinterpolation [31] is an orthogonal projection on $\mathbb{P}_n$ with respect to an absolutely continuous measure $d\mu = w(P) dP$, where the scalar products in $L_{d\mu}^2$ are discretized by a cubature rule with exactness degree $2n$, that is a discretized truncated expansion by an orthonormal polynomial basis. Such a basis is here the *orthonormal Chebyshev basis* of a cartesian bounding box $B \supseteq E$ with center (c_1, c_2) and side half-lengths ℓ_1 and ℓ_2

$$\begin{aligned} \{g_\alpha(x, y)\} &= \left\{ \frac{1}{\sqrt{\ell_1 \ell_2}} \hat{f}_\alpha(x, y) \right\}, \quad \alpha = (\alpha_1, \alpha_2), \quad 0 \leq \alpha_1 + \alpha_2 \leq n, \\ \hat{f}_\alpha(x, y) &= c_{\alpha_1} c_{\alpha_2} T_{\alpha_1} \left(\frac{x - c_1}{\ell_1} \right) T_{\alpha_2} \left(\frac{y - c_2}{\ell_2} \right), \quad (x, y) \in B, \end{aligned} \quad (13)$$

where c_s is the normalization constant for the orthogonal Chebyshev basis $\{T_s\}$ in $[-1, 1]$. On the other hand, the cubature rule is given by the positive weights $\{u_s\}$ (scaled by $\ell_1 \ell_2$) and affinely mapped nodes

$\{(\xi_s, \eta_s)\}$ of the *near-minimal Morrow-Patterson-Xu rule* for the Chebyshev measure of the first kind in the reference square $[-1, 1]^2$, with *polynomial exactness degree* $2n$ and cardinality

$$M = (n+1)(n+3)/2, \text{ } n \text{ odd}, \quad M = (n+2)^2/2, \text{ } n \text{ even}, \quad (14)$$

cf. [8, 44] and [32].

The method can be then implemented as follows:

- compute the Morrow-Patterson-Xu weights $\{u_s\}$ and nodes $\{(\xi_s, \eta_s)\} \subset [-1, 1]^2$, $1 \leq s \leq M$ (cf. [32])
- compute the Chebyshev-Vandermonde matrix $\hat{V} = (c_{\alpha_1} c_{\alpha_2} T_{\alpha_1}(\xi_s) T_{\alpha_2}(\eta_s))_{s,\alpha} \in \mathbb{R}^{M \times N}$

then, *for every element* E

- make the first three steps (8)-(10) of Spline-Tchakaloff to compute the moments $\{\hat{m}_\alpha\} = \{c_{\alpha_1} c_{\alpha_2} m_\alpha\}$;
- compute the M nodes $\{P_s\} = \{(\ell_1 \xi_s + c_1, \ell_2 \eta_s + c_2)\} \subset B$ and weights $\{w_s\} = \text{diag}(u_s) \hat{V} \{\hat{m}_\alpha\}$ of the final cubature rule.

The proof that the rule is exact in $\mathbb{P}_n$ is an immediate consequence of the fact that $\hat{V} = (\hat{f}_\alpha(P_s))_{s,\alpha}$ and that the matrix $\text{diag}(\sqrt{u_s}) \hat{V}$ is *orthogonal*, and thus

$$(\hat{f}_\alpha(P_s))^t \{w_s\} = \hat{V}^t \{w_s\} = \hat{V}^t \text{diag}(u_s) \hat{V} \{\hat{m}_\alpha\} = \{\hat{m}_\alpha\}, \quad (15)$$

i.e. the rule is exact on the polynomial basis $\{\hat{f}_\alpha(x, y)\}$.

Notice that the first two steps are *element independent* and can be done once and for all. The bulk of the algorithm is now in the moment computation, because then the weights are obtained by a *single matrix-vector product* (with fixed matrix of relatively small size, since $M \approx N = (n+1)(n+2)/2$), followed by a componentwise scaling of the resulting vector. Moreover, since *no matrix factorization* and *no system solving* is performed, we completely *avoid conditioning issues*. In addition, as proved in [41] not only the stability parameter σ in (2) is bounded but also $\sigma \rightarrow 1$ as $n \rightarrow \infty$, so that the cubature rule is *almost optimally stable* even with some negative weights.

Such a combination of *no conditioning problem*, *low construction cost*, *stability* and *low cardinality*, make Spline-Chebyshev a valid competitor with the method proposed in [17] and a good choice in FEM and VEM computations, with a main drawback: like [17] the method relies on the possibility of sampling in a bounding box, i.e. also outside the element. This feature, shared by Spline-Gauss in nonconvex instances, is not a problem for PDEs with smooth coefficients. On the other hand, it becomes a limitation in cases where one may have coefficients which are jumping between elements, or are not even defined outside the computational domain where the PDE is to be numerically approximated. This is the reason that has led to the fourth method, Spline-Fekete, described in the next subsection.

2.4. Spline-Fekete

The idea of constructing cubature rules by polynomial interpolation at Fekete-like points of complex-shaped integration domains, is not new (cf. e.g. [14]), but is here codified for the first time in the application to curved elements with spline boundary. As observed above, the main motivation is to provide implementers of curved polytopal FEM/VEM with algebraic cubature rules that have *low-cardinality internal nodal sets* and *acceptable stability features*, at a *low/moderate computational cost*.

We recall that Fekete-like sets (also called discrete extremal sets) are finite point distributions on a compact set that (*nearly*) *maximize* the absolute value of the *Vandermonde determinant*, giving good stability and convergence properties to polynomial interpolation (Lebesgue constants with $\mathcal{O}(N)$ growth, but often much lower in practice) and to the correspondent interpolatory cubature. Algorithms for their extraction from polynomial meshes, that are sequences of low-cardinality finite norming sets for the uniform norm on polynomial spaces, have been deeply studied over the last two decades. One of the most relevant properties

is that the discrete uniform probability measure supported at such discrete extremal sets converges (weakly) to the pluripotential theoretic equilibrium measure of the compact set; cf. e.g. [5, 6, 37].

We focus here on *approximate Fekete points* of an element E , which can be computed by QR factorization with column pivoting of a transposed Vandermonde matrix at a suitable discretization of E . The Spline-Fekete algorithm can be summarized as follows:

- compute the moments $\{m_\alpha\}$ and the Halton sets $H_k = \{Z_\ell\}_{1 \leq \ell \leq M_k} \subset E$ as in Spline-Tchakaloff;
- compute the transposed Chebyshev-Vandermonde matrices $V_k^t = (f_\alpha(Z_\ell))_{\alpha, \ell} \in \mathbb{R}^{N \times M_k}$ and solve the underdetermined moment-matching system

$$V_k^t u_k = \{m_\alpha\}, \quad k = 0, 1, 2, \dots \quad (16)$$

by QR factorization with column pivoting [7] (i.e. directly by the backslash operator in Matlab), until the stability parameter $\sigma = \sum_\ell |u_k(\ell)| / \text{area}(E)$ goes below a given threshold for a suitable k^* ;

- select the $M = N$ nonzero weights $\{w_s\} = \{u_{k^*}(\ell) \neq 0\}$ and nodes $\{P_s\} = \{Z_\ell : u_{k^*}(\ell) \neq 0\}$ of the final cubature formula.

We stress that the resulting nonzero weights are not all positive in general, but in all our numerical tests the negative ones turn out to be few and of small size. Though we have not a theoretical proof of this fact, but only the qualitative explanation that Fekete-like points are near-optimally distributed for polynomial interpolation, the final cubature rule has good stability properties for the degree range considered, with a stability parameter σ staying not far from the optimal value 1.

3. Numerical examples

In this section we construct and test, for the purpose of illustration, the four cubature rules described above on two elements with spline boundary. The first is convex and has seven sides (cf. Figure 1), six linear and one curved, while the second is nonconvex (cf. Figure 4) and has five sides, four linear and one curved with a complicated shape. Both can be thought as the intersection of convex polygonal elements with a curved boundary. The curved sides are tracked by cubic splines with ten control points on the convex arc, seven control points on the concave arc, and not-a-knot additional constraints. All the tests have been performed by Matlab v2024b on an Intel Core Ultra 5 125H processor.

The distribution of cubature nodes at exactness degree $n = 10$ is displayed in Figures 1 and 2. Notice that by construction Spline-Tchakaloff, Spline-Fekete, and Spline-Gauss in the convex case, have internal nodes, whereas Spline-Chebyshev has by construction also external nodes and Spline-Gauss can have external nodes in nonconvex instances. The distribution of the corresponding weights are displayed in Figures 2 and 5, respectively. We recall that by construction Spline-Tchakaloff, and Spline-Gauss in the convex case, have positive weights, whereas negative weights (but few and of small size) occur with Spline-Chebyshev, Spline-Fekete (where they turn out to be located at the boundary), and can occur for Spline-Gauss in nonconvex instances (unless it is possible to choose a suitable reference line as described in [36]).

The essential parameters of the four cubature rules are reported in Tables 1-2 and Figures 3 and 6. The figures show the distribution and the geometric mean of the relative integration errors for 100 trials a family random polynomials with degrees $n = 2, 4, 6, \dots, 16$. Such geometric means deviate of at most two orders of magnitude with respect to machine precision.

In Tables 1-2, at the same sequence of exactness degrees, we display the rule cardinality, the stability parameter $\sigma = \sum_s |w_s| / \text{area}(E)$, and the CPU time. By construction, Spline-Tchakaloff and Spline-Fekete have a cardinality equal to $N = \dim(\mathbb{P}_n) = (n+1)(n+2)/2$. On the other hand the cardinality of Spline-Chebyshev is only a little higher than N , being that of the Morrow-Patterson-Xu rule of a bounding box as in (14), whereas the cardinality of Spline-Gauss is much bigger, cf. (7). The computing time is of the order of 10^{-4} seconds for Spline-Gauss and Spline-Chebyshev at all degrees, whereas for Spline-Tchakaloff and Spline-Fekete varies from 10^{-3} to 10^{-2} seconds (the latter being in any case at least 3-4 times faster).

We can conclude by the following observations on possible selection of the rule among the four presented:

- all the rules, even those with some negative weights, show excellent stability properties in the degree range considered (suitable for FEM and VEM), with a stability parameter equal or very close to 1;
- if computing time is the key parameter, as in large scale applications, Spline-Gauss and Spline-Chebyshev are suitable choices, with a drawback of the former concerning large cardinality of the support (that could affect the construction time of stiffness and mass matrices), especially when the same finite element discretization has to be used for different equations on a given domain (e.g. for parametric or time-dependent problems);
- if only internal nodes are desired, irrespectively of the element shape, then one should choose either Spline-Tchakaloff or Spline-Fekete (faster), for example in cases where one may have coefficients which are jumping between elements, or are not even defined outside of the computational domain where the PDE is to be numerically approximated;
- if positivity of the weights is the desired feature, again irrespectively of the element shape, then Spline-Tchakaloff and Spline-Chebyshev are the choices, with the former guaranteeing also positive weights and the latter being the fastest method.

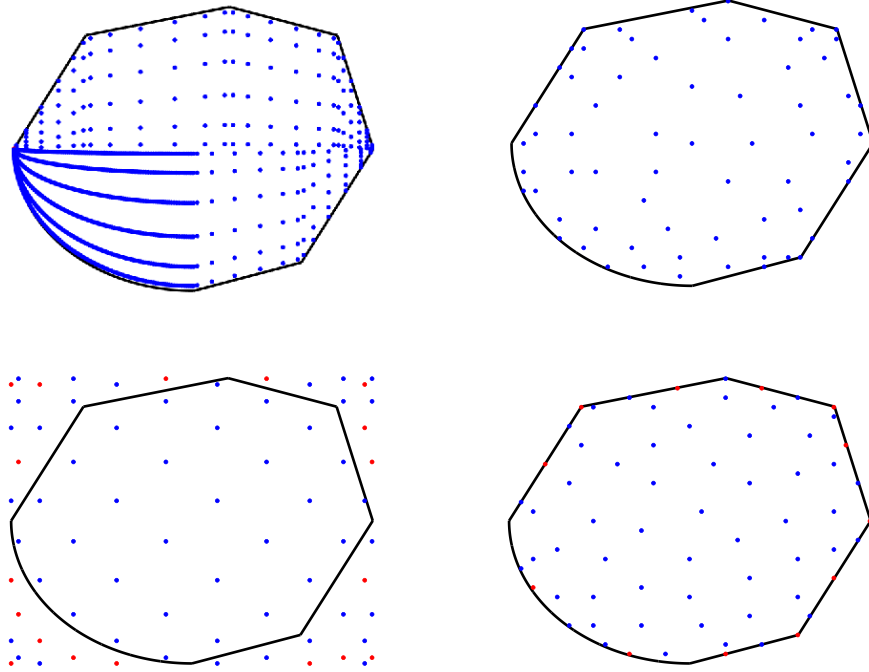


Figure 1: Nodes of the four rules with polynomial degree of exactness $n = 10$ on a convex spline element E : Spline-Gauss (top-left, 684 nodes), Spline-Tchakaloff (top-right, 66 nodes), Spline-Chebyshev (bottom-left, 72 nodes), Spline-Fekete (bottom-right, 66 nodes). Blue dots: nodes with positive weights; red dots: nodes with negative weights.

Acknowledgements.

Work partially supported by the DOR funds of the University of Padova. This research has been accomplished within the Community of Practice “Green Computing” of the Arqus European University Alliance, towards sustainable computing for massive simulation and design by finite elements, and within the INdAM-GNCS and SIMAI Italian research communities.

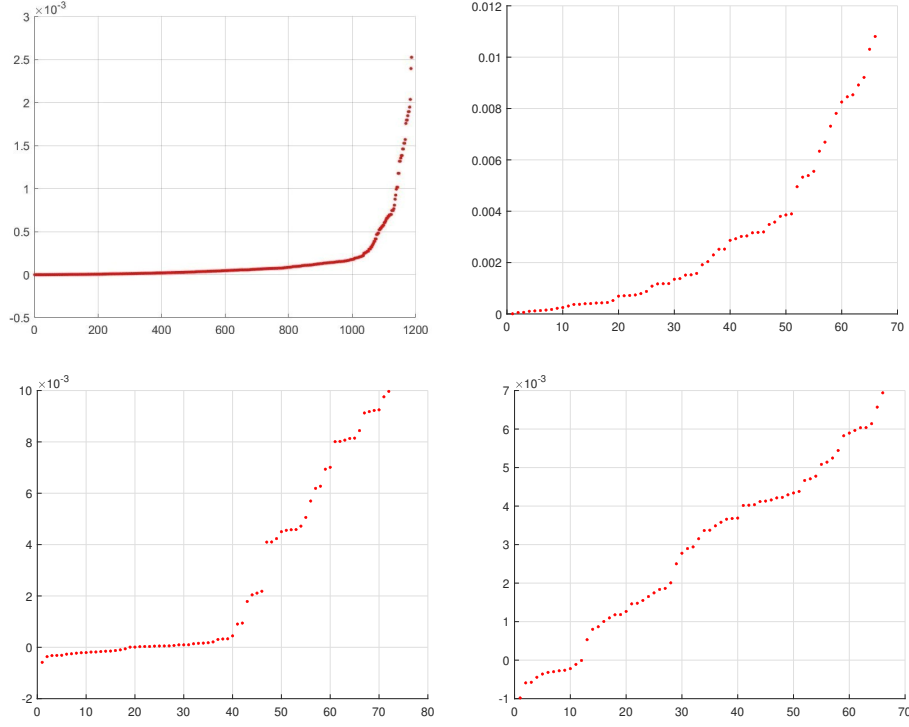


Figure 2: Weights magnitude distribution of the four rules with polynomial degree of exactness $n = 10$ on the convex spline element E of Figure 1: Spline-Gauss (top-left), Spline-Tchakaloff (top-right), Spline-Chebyshev (bottom-left), Spline-Fekete (bottom-right).

References

- [1] P.F. Antonietti, P. Houston, G. Pennesi, *Fast numerical integration on polytopic meshes with applications to discontinuous Galerkin finite element methods*, J. Sci. Comput. 77 (2018) 339–1370.
- [2] T.M. Apostol, *Calculus, Volume 2: Multi-variable calculus and linear algebra with applications to differential equations and probability (2nd ed.)*, Wiley, 1969.
- [3] E. Artioli, L. Beirão da Veiga, F. Dassi, *Curvilinear Virtual Elements for 2D solid mechanics applications*, Comput. Meth. Appl. Mech. Eng. 359 (2020).
- [4] E. Artioli, A. Sommariva and M. Vianello, *Algebraic cubature on polygonal elements with a circular edge*, Comput. Math. Appl. 79 (2020), 2057–2066.
- [5] L. Bos, J.P. Calvi, N. Levenberg, A. Sommariva, M. Vianello, *Geometric Weakly Admissible Meshes, Discrete Least Squares Approximations and Approximate Fekete Points*, Math. Comp. 80 (2011), 1601–1621.
- [6] L. Bos, S. De Marchi, A. Sommariva, M. Vianello, *Computing multivariate Fekete and Leja points by numerical linear algebra*, SIAM J. Numer. Anal. 48 (2010), 1984–1999.
- [7] P.A. Businger and G.H. Golub, *Linear least-squares solutions by Householder transformations*, Numer. Math. 7 (1965), 269–276.
- [8] M. Caliarì, S. De Marchi, R. Montagna, M. Vianello, *Hyper2d: a numerical code for hyperinterpolation on rectangles*, Appl. Math. Comput. 183 (2006), 1138–1147.

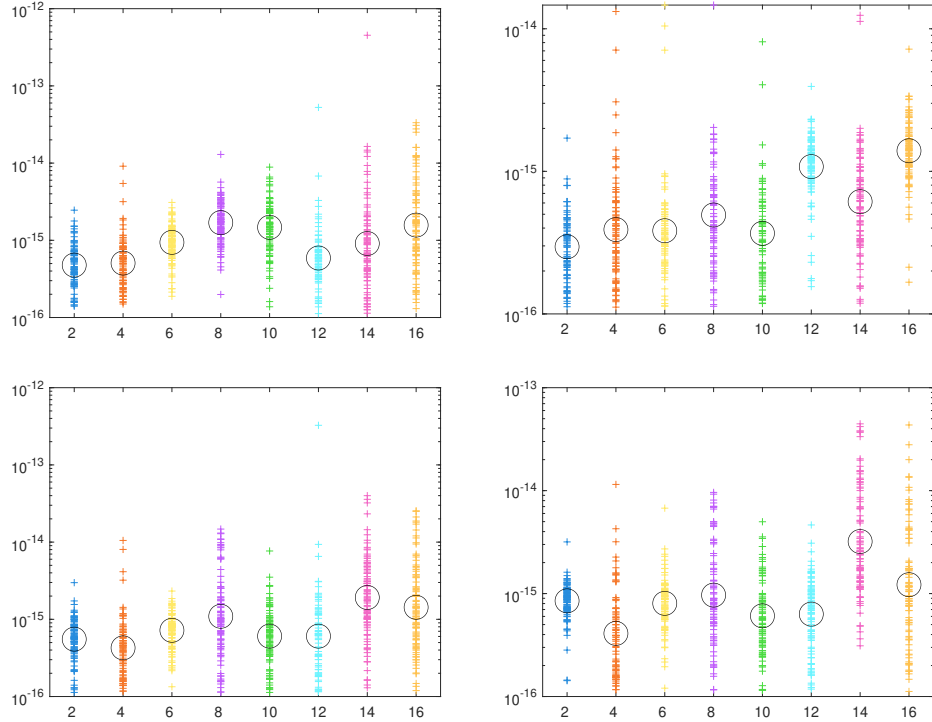


Figure 3: Geometric means of the integration relative errors for 100 trials of random polynomials $(c_0 + c_1x + c_2y)^n$ on the convex spline element E of Figure 1: Spline-Gauss (top-left), Spline-Tchakaloff (top-right), Spline-Chebyshev (bottom-left), Spline-Fekete (bottom-right).

- [9] E.B. Chin, J.B. Lasserre, N. Sukumar, *Numerical integration of homogeneous functions on convex and nonconvex polygons and polyhedra*, Comput. Mech. 56 (2015), 967–981.
- [10] E. B. Chin, N. Sukumar, *Scaled boundary cubature scheme for numerical integration over planar regions with affine and curved boundaries*, CMAME 380 (2021), 110066.
- [11] F. Dassi, Vem++, a C++ library to handle and play with the virtual element method, Numer. Algorithms 101 (3) (2026), 1633–1675.
- [12] M. Dell’Orto, M. Dessole, F. Marcuzzi, *The Lawson-Hanson Algorithm with Deviation Maximization: Finite Convergence and Sparse Recovery*, Numer. Linear Algebra Appl. 30(5) (2023).
- [13] M. Dessole, F. Marcuzzi, *Deviation maximization for rank-revealing QR factorizations*, Numer. Algorithms 91 (2022), 1047–1079.
- [14] M. Gentile, A. Sommariva, M. Vianello, Polynomial interpolation and cubature over polygons, J. Comput. Appl. Math. 235 (2011), 5232–5239.
- [15] D. Gunderman, K. Weiss, J.A. Evans, *Spectral mesh-free quadrature for planar regions bounded by rational parametric curves*, Comput. Aided Geom. Des. 130, 102944.
- [16] K. S. Kirilov, J. Zhou, J. Peiró, M. Green, D. Moxey, L. Beirão da Veiga, A. Russo, F. Dassi, Mesh generation of curvilinear polygons for the high-order virtual element method (VEM), Comput. Aided Des. 190 (2026), 103966.
- [17] C. Langlois, T. van Putten, H. Bériot, E. Deckers, *Frugal numerical integration scheme for polytopal domains*, Eng. Comput. (2024).

	Spline-Gauss			Spline-Tchakaloff			Spline-Chebyshev			Spline-Fekete		
deg n	card	stab	cpu	card	stab	cpu	card	stab	cpu	card	stab	cpu
2	132	1.00	7.5e-4	6	1.00	2.0e-3	8	1.21	5.5e-4	6	1.03	2.0e-3
4	297	1.00	7.1e-4	15	1.00	2.2e-3	18	1.08	6.8e-4	15	1.10	2.1e-3
6	528	1.00	8.4e-4	28	1.00	3.9e-3	32	1.06	6.3e-4	28	1.00	3.5e-3
8	825	1.00	7.2e-4	45	1.00	6.1e-3	50	1.06	6.3e-4	45	1.05	2.9e-3
10	1188	1.00	7.6e-4	66	1.00	1.3e-2	72	1.04	6.9e-4	66	1.05	4.6e-3
12	1617	1.00	7.8e-4	91	1.00	3.2e-2	98	1.05	7.5e-4	91	1.01	8.0e-3
14	2112	1.00	8.3e-4	120	1.00	5.1e-2	128	1.04	8.2e-4	120	1.02	1.5e-2
16	2673	1.00	1.0e-3	153	1.00	6.8e-2	162	1.04	8.5e-4	153	1.06	2.6e-2

Table 1: Essential parameters of the four rules on the convex spline element E of Figure 1; rule cardinality, the stability parameter $\sigma = \sum_s |w_s|/area(E)$, cputime in seconds.

	Spline-Gauss			Spline-Tchakaloff			Spline-Chebyshev			Spline-Fekete		
deg n	card	stab	cpu	card	stab	cpu	card	stab	cpu	card	stab	cpu
2	76	1.16	7.9e-4	6	1.00	1.5e-3	8	1.24	4.2e-4	6	1.00	1.4e-3
4	171	1.16	7.8e-4	15	1.00	1.7e-3	18	1.20	4.9e-4	15	1.16	1.5e-3
6	304	1.16	7.7e-4	28	1.00	2.6e-3	32	1.15	4.4e-4	28	1.05	1.8e-3
8	475	1.16	7.7e-4	45	1.00	1.5e-2	50	1.10	4.6e-4	45	1.17	3.1e-3
10	684	1.16	8.1e-4	66	1.00	1.3e-2	72	1.13	4.8e-4	66	1.04	3.4e-3
12	931	1.16	8.7e-4	91	1.00	4.0e-2	98	1.08	5.5e-4	91	1.10	7.2e-3
14	1216	1.16	8.7e-4	120	1.00	6.9e-2	128	1.08	6.7e-4	120	1.22	1.3e-2
16	1539	1.16	9.7e-4	153	1.00	8.5e-2	162	1.09	6.5e-4	153	1.13	2.1e-2

Table 2: Essential parameters of the four rules on the concave spline element E of Figure 4; rule cardinality, the stability parameter $\sigma = \sum_s |w_s|/area(E)$, cputime in seconds.

- [18] C.L. Lawson and R.J. Hanson, *Solving least squares problems*, Revised reprint of the 1974 original. Classics in Applied Mathematics 15, SIAM, Philadelphia, 1995.
- [19] C. Litterer, T. Lyons, *High order recombination and an application to cubature on Wiener space*, Ann. Appl. Probab. 22 (2012), 1301–1327.
- [20] J.C. Mason, D.C. Handscombe, *Chebyshev Polynomials*, Chapman&Hall, 2002.
- [21] S. Mehrotra, D. Papp, *Generating moment matching scenarios using optimization techniques*, SIAM J. Optim. 23 (2013), 963–999.
- [22] M. Montanari, G.M. Santi, R. Sevilla, A. Liverani, N. Petrinic, *NURBS-enhanced finite element method (NEFEM) on quadrilateral meshes*, Finite Elem. Anal. Des. 231 (2024), 104099.
- [23] C.R. Morrow, T.N.L. Patterson, *Construction of algebraic cubature rules using polynomial ideal theory*, SIAM J. Numer. Anal. 15 (1978), 953–976.
- [24] S.E. Mousavi, H. Xiao, N. Sukumar, *Generalized Gaussian quadrature rules on arbitrary polygons*, Int. J. Numer. Meth. Eng. 82 (2010), 99–113.
- [25] F. Piazzon, A. Sommariva, M. Vianello, *Caratheodory-Tchakaloff Subsampling*, Dolomites Res. Notes Approx. DRNA 10 (2017), 5–14.
- [26] L. Rinaldi, *Cheap integration routines in Python*, <https://github.com/laura-rinaldi/>.
- [27] L. Rinaldi, A. Sommariva, M. Vianello, *Effective numerical integration on complex shaped elements by discrete signed measures*, Finite Elem. Anal. Des. 254 (2026), 104505.

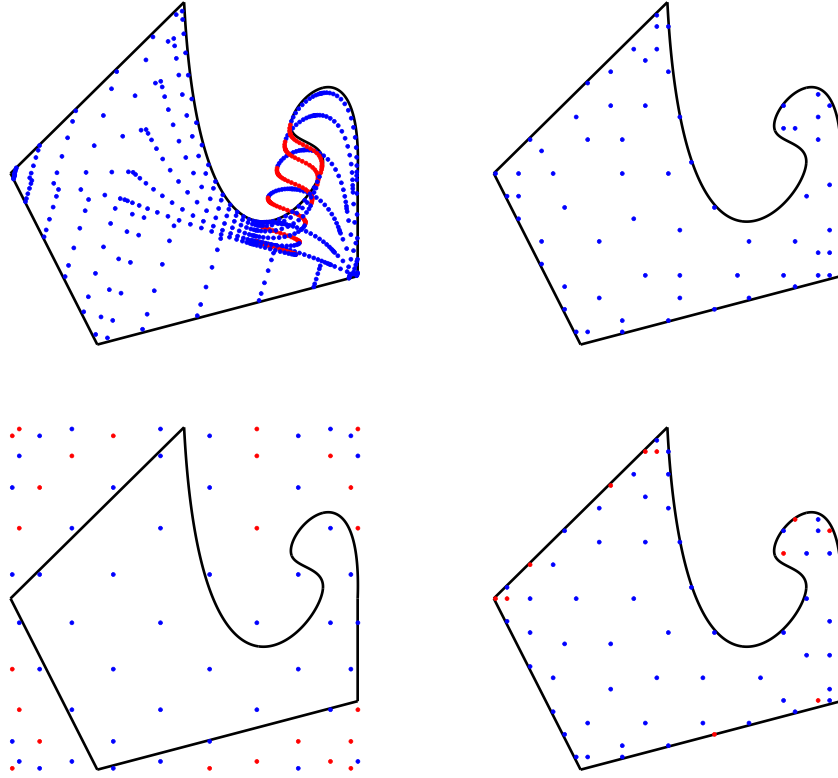


Figure 4: Nodes of the four rules with polynomial degree of exactness $n = 10$ on a concave spline element E : Spline-Gauss (top-left, 684 nodes), Spline-Tchakaloff (top-right, 66 nodes), Spline-Chebyshev (bottom-left, 72 nodes), Spline-Fekete (bottom-right, 66 nodes). Blue dots: nodes with positive weights; red dots: nodes with negative weights.

- [28] L. Rinaldi, A. Sommariva, M. Vianello, *ORTHOCUB: integral and differential cubature rules by orthogonal moments*, JAS - J. Approx. Software 3, 2 (2026).
- [29] E.K. Ryu, S.P. Boyd, *Extensions of Gauss quadrature via linear programming*, Found. Comput. Math. 15 (2015), 953–971.
- [30] R. Sevilla, S. Fernández-Méndez, A. Huerta, *Comparison of high-order curved finite elements*, Int. J. Numer. Meth. Eng. 87 (2011), 719–734.
- [31] I.H. Sloan, *Interpolation and Hyperinterpolation over General regions*, J. Approx. Theory 83 (1995), 238–254.
- [32] A. Sommariva, *Collection of minimal and almost minimal rules in the square and the cube (with Matlab/Octave codes)*: <https://sites.google.com/view/alvisesommarivaunipd/home-page/sets>.
- [33] A. Sommariva, *Cheap integration routines in Matlab*, <https://github.com/alvisesommariva/>.
- [34] A. Sommariva, *CUB_curvilinear_splines*: Matlab codes for algebraic cubature on spline curvilinear elements (2026) https://codeberg.org/alvisesommariva/CUB_curvilinear_splines/.
- [35] A. Sommariva, M. Vianello, *Product Gauss cubature over polygons based on Green’s integration formula*, BIT Numerical Mathematics 47 (2007), 441–453.
- [36] A. Sommariva, M. Vianello, *Gauss-Green cubature and moment computation over arbitrary geometries*, J. Comput. Appl. Math. 231 (2009), 886–896.

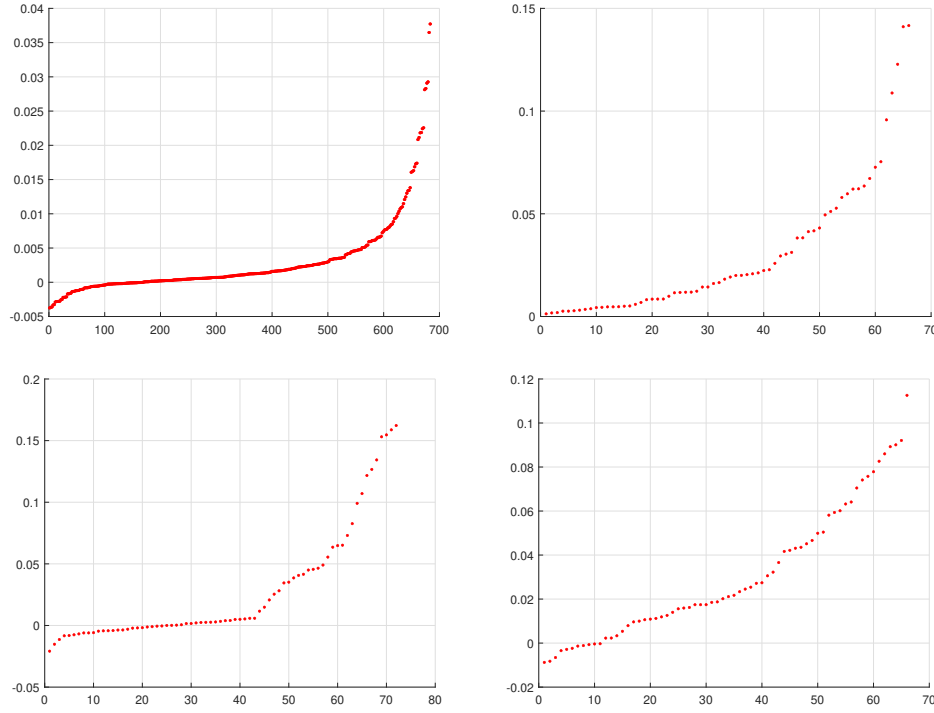


Figure 5: Weights magnitude distribution of the four rules with polynomial degree of exactness $n = 10$ on the concave spline element E of Figure 4: Spline-Gauss (top-left), Spline-Tchakaloff (top-right), Spline-Chebyshev (bottom-left), Spline-Fekete (bottom-right).

- [37] A. Sommariva, M. Vianello, *Computing approximate Fekete points by QR factorizations of Vandermonde matrices*, Comput. Math. Appl. 57 (2009), 1324–1336.
- [38] A. Sommariva, M. Vianello, *Compression of multivariate discrete measures and applications*, Numer. Funct. Anal. Optim. 36 (2015) 1198–1223.
- [39] A. Sommariva, M. Vianello, *Computing Tchakaloff-like cubature rules on spline curvilinear polygons*, Dolomites Res. Notes Approx. DRNA 14 (2021), 1–11.
- [40] A. Sommariva, M. Vianello, *TetraFreeQ: tetrahedra-free quadrature on polyhedral elements*, Appl. Numer. Math. 200 (2024), 389–398.
- [41] A. Sommariva, M. Vianello, *Cheap and stable quadrature on polyhedral elements*, Finite Elem. Anal. Des. 251 (2025), 104409.
- [42] V. Tchakaloff, *Formules de cubatures mécaniques à coefficients non négatifs* (French), Bull. Sci. Math. 81 (1957), 123–134.
- [43] M. Tchernychova, *“Caratheodory” cubature measures*, Ph.D. dissertation in Mathematics (supervisor: T. Lyons), University of Oxford, 2015, available at <https://ora.ox.ac.uk/objects/uuid:a3a10980-d35d-467b-b3c0-d10d2e491f2d>.
- [44] Y. Xu, *Minimal Cubature Rules: Theory and Practice*, Cambridge Monographs on Applied and Computational Mathematics 45, Cambridge Univ. Press, 2025.
- [45] D.R. Wilhelmsen, *A Nearest Point Algorithm for Convex Polyhedral Cones and Applications to Positive Linear approximation*, Math. Comp. 30 (1976), 48–57.

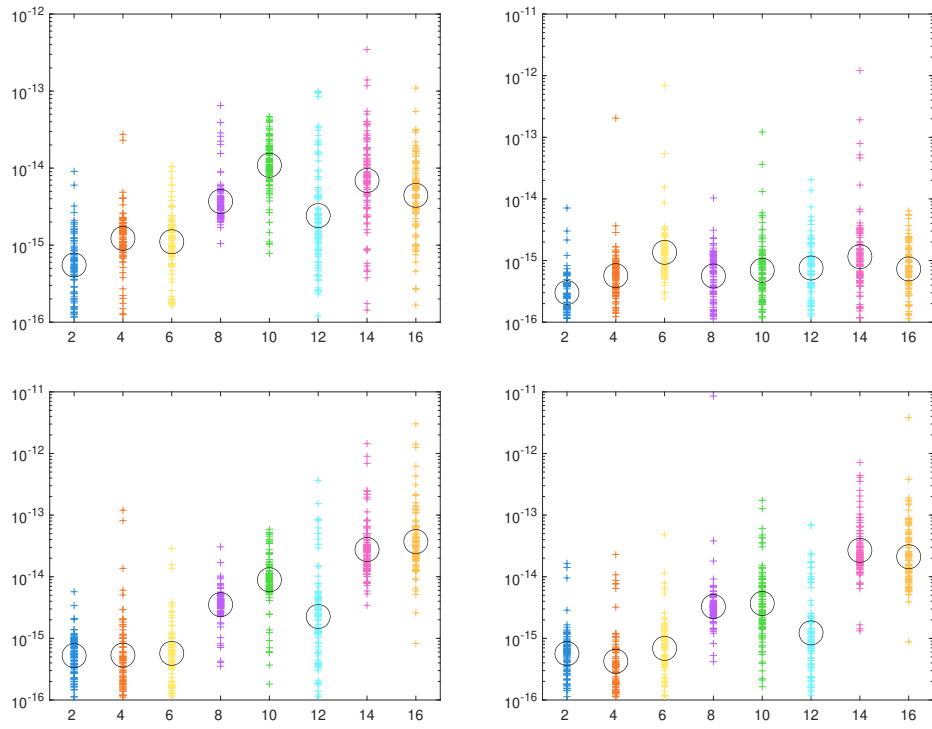


Figure 6: Geometric means of the integration relative errors for 100 trials of random polynomials $(c_0 + c_1x + c_2y)^n$ on the concave spline element E of Figure 4: Spline-Gauss (top-left), Spline-Tchakaloff (top-right), Spline-Chebyshev (bottom-left), Spline-Fekete (bottom-right).