Computing Tchakaloff-like cubature rules on spline curvilinear polygons *

A. Sommariva[1] and M. Vianello
Department of Mathematics, University of Padova, Italy

September 11, 2020

2010 AMS subject classification: Primary 65D32; Secondary: 65D07.

Keywords: low-cardinality algebraic cubature, positive interior rules, spline curvilinear polygons, crossing number, winding number, Tchakaloff sets, underdetermined moment systems, nonnegative least squares.

Abstract

We present an algorithm that computes a PI-type (Positive Interior) algebraic cubature rule of degree n with at most $(n+1)(n+2)/2$ nodes, over spline curvilinear polygons. The key ingredients are a theorem by Davis on *Tchakaloff discretization sets*, a specific *in-domain* algorithm for such spline polygons and the sparse nonnegative solution of underdetermined moment matching systems by the Lawson-Hanson NonNegative Least Squares solver. A numerical code (implemented in Matlab) is also provided, together with several numerical tests.

1 Introduction

During the last years, there has been an increasing interest on algebraic cubature rules over 2d polygonal and curvilinear domains, especially in the framework of the numerical solution of PDEs. To quote, among many others, a couple of relevant issues, we may recall that within Virtual Elements Methods (VEM), there is need for low-cardinality formulae on elements that have

*Work partially supported by the DOR funds and the biennial project BIRD 192932 of the University of Padova, and by the GNCS-INdAM. This research has been accomplished within the RITA "Research ITalian network on Approximation".

[1]corresponding author: alvise@math.unipd.it

nonstandard shape with possibly curved edges, that could be typically well approximated by parametric splines (see e.g. [1, 3, 4, 6] and the references therein). In these instances, the required Algebraic Degree of Precision, a term often shortened by the acronym ADE, is usually low (typically from 2 to 4), but rules of PI-type, i.e. with *Positive weights* and *Interior nodes*, are in many senses more appealing.

On the other hand, the importance of having algebraic cubature rules on general planar domains with curved boundaries, possibly tracked by splines, has been shown for example also in the framework of Lagrangian flux calculation through a fixed curve for scalar conservation laws, cf. [27, 28].

In [17, 20] we already coped the problem of determining algebraic cubature rules on general domains with curved boundaries, approximated by polynomial or piecewise polynomial parametric curves. These rules turn out to be of PI-type only in convex or special nonconvex regions. When such formulae are available, their cardinality can be reduced by a general method for the compression of discrete measures, cf. e.g. [21].

Moreover, we have recently introduced in [5] an algorithm that determines a formula of PI-type and prescribed degree of precision, even on polygons with complicated shapes, that can be nonconvex, not simple and not simply connected. In particular, if the cardinality of the rule is an issue, we were able to provide rules of PI-type, $ADE = n$ and a number of nodes at most $N = N_n = \dim(\mathbb{P}_n^2) = (n+1)(n+2)/2$ by means of a numerical implementation of Tchakaloff theorem (see [21, 25]). The corresponding Matlab codes make use of the recent `polyshape` environment and can be retrieved at [18].

The spline curvilinear counterpart is delicate and we propose here a routine that fills the gap, providing a low-cardinality rule of PI-type not only over polygons (corresponding to linear splines), but also over Jordan domains with spline curvilinear boundary (spline curvilinear polygons).

A key tool is a specific in-domain routine that works on any spline curvilinear polygon, say S, by which the algorithm determines a grid-based discretization $X = \{P_i\} \subset \mathcal{S}$. Then it evaluates the Vandermonde-like matrix $V = V_n(X) = (\phi_j(P_i))_{i,j}$ w.r.t. a certain polynomial basis $\{\phi_j\}$ of total degree n over $\mathcal{S}$. Finally, it computes the moments $\gamma_j = \int_{\mathcal{S}} \phi_j(x,y)\,dx\,dy$ by Gauss-Green formula and univariate Gaussian quadrature, and resorts to a numerical implementation of Tchakaloff theorem, seeking a sparse nonnegative solution of the underdetermined system $V^T u = \gamma$ where the vector of the weights u has $\nu \leq N = (n+1)(n+2)/2$ nonzero components, that determine the weights and nodes of a PI-type cubature rule, say $\{(w_j, Q_j)\}_{j=1}^{\nu}$, $\{Q_j\} \subset X$. The existence of such a solution relies on a classical and substantially overlooked general result on "*Tchakaloff (discretization) sets*", proved in [26] generalizing a result by Davis [8].

The paper is organized as follows. In section 2, we introduce our *in-domain* routine, that works not only on linear polygons but even on spline curvilinear polygons $\mathcal{S}$. In section 3 we briefly recall some relevant results about Tchakaloff theorem and its numerical implementation. In section 4 we explain our new method for determining Tchakaloff-like cubature rules on $\mathcal{S}$, pointing out how to compute the required moments $\{\gamma_j\}$ and how to use the fundamental results of [26]. Finally in section 5 we show some numerical results on two nonconvex domains, concerning both, the *in-domain* procedure and the final low-cardinality PI-type cubature rule.

2 In-domain for spline curvilinear polygons

Let $\mathcal{S} \subset \mathbb{R}^2$ be a Jordan domain and suppose that its boundary $\partial\mathcal{S}$

1. can be described by parametric equations $x = \tilde{x}(t)$, $y = \tilde{y}(t)$, $t \in [a, b]$, $\tilde{x}, \tilde{y} \in C([a, b])$, $\tilde{x}(a) = \tilde{x}(b)$ and $\tilde{y}(a) = \tilde{y}(b)$;

2. there is a partition $\{I^{(k)}\}_{k=1,\dots,M}$ of $[a, b]$, and partitions $\{I_j^{(k)}\}_{j=1,\dots,m_k}$ of each $I^{(k)}$, such that the restrictions of $\tilde{x}$, $\tilde{y}$ on each $I^{(k)}$ are splines of degree δ_k, w.r.t. the subintervals $\{I_j^{(k)}\}_{j=1,\dots,m_k}$.

Typical instances are *approximations* of more general Jordan domains Ω by means of spline curvilinear regions $\mathcal{S}$. To this purpose, given the *vertices* $V_k \in \partial\Omega$, $k = 1, \dots, M + 1$, then $\partial\Omega \approx \partial\mathcal{S} := \cup_{k=1}^{M} V_k \frown V_{k+1}$ where each curvilinear side $V_k \frown V_{k+1}$ is tracked by a spline curve of degree δ_k, interpolating an ordered subsequence of control points $P_{1,k} = V_k, P_{2,k}, \dots, P_{m_k-1,k}, P_{m_k,k} = V_{k+1}$ with a suitable parametrization that determines each $I_j^{(k)}$ (and thus each $I^{(k)}$).

In this section we introduce an algorithm that establishes if a point $P \in \mathbb{R}^2$ is inside, outside or on the boundary of $\partial\mathcal{S}$.

The key point is the well-known *Jordan curve theorem* that states that a point P belongs to a Jordan domain Ω if and only if, having taken a point $P^* \notin \Omega$ then the segment $\overline{P^*P}$ crosses $\partial\Omega$ an odd number $c(P)$ of times.

Algorithms of this type are popular in the polygonal case, where already the determination of the *crossing number* may be not straightforward. In fact, when $\overline{P^*P}$ crosses a vertex of $\partial\Omega$, there are cases in which difficulties may arise. Much worse, the segment $\overline{P^*P}$ may include a segment of the

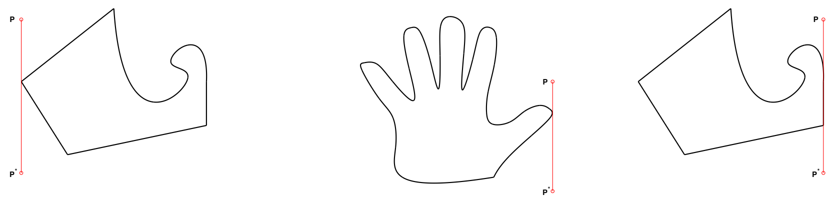

Figure 1: Critical situations for establishing the crossing number on spline curvilinear polygons.

polygon and the definition of $c(P)$ is not trivial. These *critical* situations are usually treated with care.

Similarly, in the more general instance of spline curvilinear domains $\mathcal{S}$, a difficult situation arises when the boundary $\partial\mathcal{S}$ has a *critical* point $S = (\tilde{x}(\gamma), \tilde{y}(\gamma))$ where

$$\lim_{t\to\gamma^-} \tilde{y}'(t) \lim_{t\to\gamma^+} \tilde{y}'(t) < 0,$$

that geometrically means that there is locally a vertical turn of boundary from left to right (or conversely from right to left).

If the in-domain analysis of the point P is performed by means of *vertical* segments $\overline{P^*P}$ with the abscissa of P equal to that of S^*, then Jordan theorem cannot be applied.

Even worse, the *vertical* segments $\overline{P^*P}$ may contain a portion of the boundary that consist of a vertical segment (see Figure 1).

These situations are not difficult to be detected, and we will suppose first that the vertical segment $\overline{P^*P}$ does not contain any *critical point* or vertical side.

Under these assumptions, let $\mathcal{R}$ be a rectangle, with sides parallel to the cartesian axis, that contains the spline curvilinear domain $\mathcal{S}$ and such that $\partial\mathcal{R} \cap \partial\mathcal{S} = \emptyset$, i.e. their boundaries do not intersect. Let P^* be the point of $\partial\mathcal{R}$ that shares the same absissa of P but has ordinate strictly inferior. We want to compute how many times the vertical segment $\overline{P^*P}$ crosses $\partial\mathcal{S}$.

The technique that we explain below consists essentially in covering the boundary $\partial\mathcal{S}$ with a region $\mathcal{B}$ defined by the union of certain rectangles whose interiors do not intersect. In particular, the portion of the boundary $\partial\mathcal{S}$ in each of these rectangles will not have any turning point and will be parametrized by two polynomials.

Finally we will show that if the point $P \in \mathcal{S}\backslash\mathcal{B}$ then the evaluation of the crossing number $c(P)$ is straightfoward, otherwise it requires the solution of some polynomial equations.

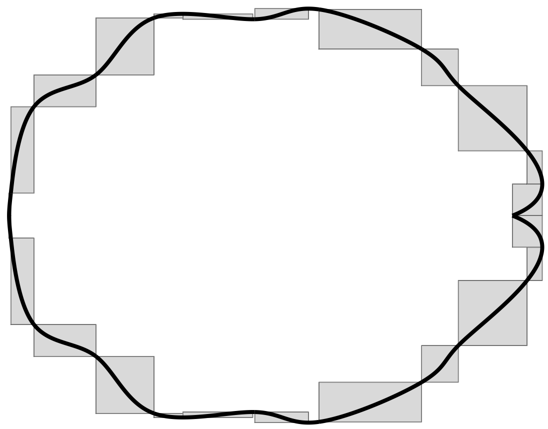

Figure 2: A domain $\mathcal{S}$ of (spline) curvilinear type and its *monotone boxes*.

To this purpose, suppose that

$$I_j^{(k)} = [t_j^{(k)}, t_{j+1}^{(k)}],\ k = 1, \dots, M,\ j = 1, \dots, m_k - 1,$$

determines a partition of the interval $[a, b]$ and that $\partial\mathcal{S} = \{(\tilde{x}(t), \tilde{y}(t)),\ t \in [a, b]\}$, where the restriction of $\tilde{x}$, $\tilde{y}$ to $I_j^{(k)}$ is a polynomial of degree δ_k.

If $\tilde{x}'$ changes sign in $(t_j^{(k)}, t_{j+1}^{(k)})$, then define as $\mathcal{N}_j^{(k)} = \{t_i^{(j,k)}\}_{i=1,\dots,l_{j,k}}$ the set of all the points $t_i^{(j,k)} \in (t_j^{(k)}, t_{j+1}^{(k)})$ such that $\tilde{x}'(t_i^{(j,k)}) = 0$ (observe that the restriction to $I^{(j,k)}$ is a polynomial of degree δ_k, hence its derivative exists), otherwise put $\mathcal{N}_j^{(k)} = \emptyset$. Let $T^{(j,k)} = \{t_j^{(k)}, t_{j+1}^{(k)}\} \cup \mathcal{N}_j^{(k)}$, where we suppose that its elements, say $T_i^{(j,k)}$, are in increasing order.

Observe that being $\tilde{x}$ a spline of degree δ_k, the determination of the set $\mathcal{N}_j^{(k)}$ requires the solution of a polynomial equation of degree $\delta_k - 1$, that can be computed explicitly when $\delta_k \leq 5$.

Next introduce the rectangles $\mathcal{B}_i^{(j,k)}$, that we will call *monotone boxes*,

$$\mathcal{B}_i^{(j,k)} := [\min_{t \in I_i^{(j,k)}} \tilde{x}(t), \max_{t \in I_i^{(j,k)}} \tilde{x}(t)] \times [\min_{t \in I_i^{(j,k)}} \tilde{y}(t), \max_{t \in I_i^{(j,k)}} \tilde{y}(t)].$$

where $I_i^{(j,k)} := [T_i^{(j,k)}, T_{i+1}^{(j,k)}]$.

Observe that by definition if $\mathcal{N}_j^{(k)} = \emptyset$, then there is only the monotone box $\mathcal{B}_1^{(j,k)}$.

Since $\tilde{y}$ restricted to $[T_i^{(j,k)}, T_{i+1}^{(j,k)}]$ is a polynomial of degree δ_k, the evaluation of

$$\min_{t \in [T_i^{(j,k)}, T_{i+1}^{(j,k)}]} \tilde{y}(t), \qquad \max_{t \in [T_i^{(j,k)}, T_{i+1}^{(j,k)}]} \tilde{y}(t)$$

can be easily determined once the derivative of the polynomial $\tilde{y}'$ is at hand, by computing its zeros in $[T_i^{(j,k)}, T_{i+1}^{(j,k)}]$ and the evaluation of $\tilde{y}$ at $T_i^{(j,k)}$ and $T_{i+1}^{(j,k)}$.

Thus, the restrictions of $\tilde{x}$, $\tilde{y}$ to each $I_i^{(j,k)} \subseteq [a, b]$ are polynomials of degree δ_k. In particular, being the restriction of $\tilde{x}$ a monotone function, there are no turning points of $\partial\mathcal{S}$ in the interior of each box $\mathcal{B}_i^{(j,k)}$.

Once the set $\mathcal{B} := \{\mathcal{B}_i^{(j,k)}\}$ is at hand, we apply the crossing theorem to determine if $P = (P_x, P_y)$ is in the domain $\mathcal{S}$.

To this purpose we the consider the monotone boxes

$$\mathcal{B}(P) = \{B = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \in \mathcal{B} : P_x \in [\alpha_1, \beta_1], P_y \geq \alpha_2\}$$

Intuitively, $\mathcal{B}(P)$ contains all the boxes $\mathcal{B}_l$ such that $\overline{P^*P} \cap \mathcal{B}_l$ is not empty, and that actually may contribute to the computation of the crossing number.

Take the generic monotone box $\mathcal{B}_l = [\alpha_1^{(l)}, \beta_1^{(l)}] \times [\alpha_2^{(l)}, \beta_2^{(l)}] \in \mathcal{B}(P)$. If $P_y > \beta_2^{(l)}$ then the point P is *over* the monotone box $\mathcal{B}_l$ and necessarily the segment $\overline{P^*P}$ crosses the boundary $\partial\mathcal{S}$ once in $\mathcal{B}_l$ and *below* P, due to the monotonicy of $\tilde{x}$ in $\mathcal{B}_l$.

Otherwise the point $P \in \mathcal{B}_l$. Let t^* be the unique solution of the polynomial equation $\tilde{x}(t) = P_x$ (notice that uniqueness comes from the local monotonicity of $\tilde{x}$).

Next,

- if $\tilde{y}(t^*) < P_y$ then the segment $\overline{P^*P}$ crosses the boundary $\partial\mathcal{S}$ once in the monotone box, *below* P;

- if $\tilde{y}(t^*) > P_y$ then the segment $\overline{P^*P}$ does not cross the boundary $\partial\mathcal{S}$ once in B, *below* P;

- if $\tilde{y}(t^*) = P_y$ then P is on the boundary $\partial\mathcal{S}$.

Thus counting all these crossings, we are *usually* able to determine if a point P is inside, outside or on the boundary of $\mathcal{S}$. We point out that if

$\delta_k \leq 4$ the solution of such polynomial equation can be computed explicitly, while when $\delta > 4$ one must use specific numerical methods.

In the cases in which the vertical segment $\overline{P^*P}$ contains a *critical point* or a vertical side, we have used an algorithm that is based on the well-known *winding* theorem to ascertain whether P belongs or not to $\mathcal{S}$.

To this purpose, we computed by a (shifted) Gauss-Legendre rule of sufficiently high degree of precision the so called *winding number* $\text{wind}(P, \tilde{x}, \tilde{y}) \in \mathbb{Z}$,

$$\text{wind}(P, \tilde{x}, \tilde{y}) := \frac{1}{b-a} \int_a^b \frac{\tilde{y}'(t)(\tilde{x}(t) - P_x) - \tilde{x}'(t)(\tilde{y}(t) - P_y)}{(\tilde{x}(t) - P_x)^2 + (\tilde{y}(t) - P_y)^2} dt.$$

If this quantity is odd then the point belongs to $\mathcal{S}$ otherwise is not inside such domain. This approach has been used in the Hormann-Agathos algorithm [10], by a clever computation of this quantity when $\mathcal{S}$ is actually a polygon.

We observe that the evaluation of $\text{wind}(P, \tilde{x}, \tilde{y})$ can be difficult when P is close to the boundary, but in general such quantity must not be computed with high precision, in view of the fact that $\text{wind}(P, \tilde{x}, \tilde{y})$ is an integer.

Remark 1 In our implementation, each monotone box $\mathcal{B}_i^{(j,k)}$ has abscissae ranging in the interval

$$\left[\min(\tilde{x}(T_i^{(j,k)}), \tilde{x}(T_{i+1}^{(j,k)})), \max(\tilde{x}(T_i^{(j,k)}), \tilde{x}(T_{i+1}^{(j,k)}))\right].$$

If we partition $[T_i^{(j,k)}, T_{i+1}^{(j,k)}]$ as $\cup_{s=1}^{n_{i,j,k}-1} [T_{i,s}^{(j,k)}, T_{i,s+1}^{(j,k)}]$ where

$$T_i^{(j,k)} = T_{i,1}^{(j,k)} < T_{i,2}^{(j,k)} < \ldots T_{i,n_{i,j,k}}^{(j,k)} = T_{i+1}^{(j,k)}$$

and define the monotone boxes

$$\mathcal{B}_{i,s}^{(j,k)} := [\min_{t \in I_{i,s}^{(j,k)}} \tilde{x}(t), \max_{t \in I_{i,s}^{(j,k)}} \tilde{x}(t)] \times [\min_{t \in I_{i,s}^{(j,k)}} \tilde{y}(t), \max_{t \in I_{i,s}^{(j,k)}} \tilde{y}(t)].$$

where $I_{i,s}^{(j,k)} = [T_{i,s}^{(j,k)}, T_{i,s+1}^{(j,k)}]$, we have again that $\cup \mathcal{B}_{i,s}^{(j,k)} \subseteq \mathcal{B}$ contains $\mathcal{S}$, but that can be a strict subset of $\mathcal{B}$.

In such a case, an inferior number of polynomial equations is needed to establish what points of X belong to $\mathcal{S}$. In spite of this, we observe that there is numerical evidence that a too high number of monotone boxes slows down the *in-domain* process.

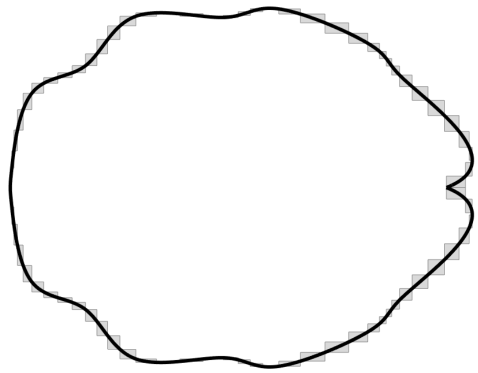

Figure 3: A domain $\mathcal{S}$ of (spline) curvilinear type and a *smaller* set of monotone boxes (compare with Figure 2).

Remark 2 In the implementation of the algorithm, to see if a point $P = (P_x, P_y)$ belongs to the boundary $\partial\mathcal{S}$, we solved a *polynomial* equation $\tilde{x}(t^*) = P_x$ and then tested that $\tilde{y}(t^*) = P_y$. In view of numerical errors, we can only establish that a point is very close to $\partial\mathcal{S}$, that is $|\tilde{y}(t^*) - P_y|$ is below a certain threshold.

Remark 3 In the case each degree δ_k is equal to 1, i.e. $\mathcal{S}$ is a polygon, it is more profitable to use to the well-known *Hormann-Agathos* algorithm, implemented in Matlab by the `inpolygon` routine.

3 Tchakaloff-like sampling

The purpose of this section is to show how to extract from a sufficiently dense discretization of a compact set a PI-type cubature rule with low cardinality (not exceeding the dimension of the exactness polynomial space). To this purpose, a fundamental existence result is Tchakaloff theorem [24], that we present in a general formulation (cf. e.g. [15]):

Theorem 1 *Let μ be a positive measure on the compact domain $D \subset \mathbb{R}^d$ and let n be a positive integer. Then there are $\nu \leq$ dim$(\mathbb{P}_n^d(D))$ points $\{Q_j\} \in D$ and positive real numbers $\{w_j\}$ such that*

$$\int_D p(P)\, d\mu = \sum_{j=1}^{\nu} w_j p(Q_j)$$

for all $p \in \mathbb{P}_n^d(D)$ (the space of d-variate polynomials of degree not exceeding n, restricted to D).

Our purpose it to determine numerically a set of such points $\{Q_j\} \subset D$, together with the corresponding positive weights. A key result was proved by Wilhelmsen in [26] (extending a result of Davis [8] to quite general functional spaces, including polynomial spaces).

Theorem 2 *Let Φ be the linear span of continuous, real-valued, linearly independent functions $\{\phi_k\}_{k=1,\dots,N}$ defined on a compact set $D \subset \mathbb{R}^d$. Assume that Φ satisfies the Krein condition (i.e. there is at least one $f \in \Phi$ which does not vanish on D) and that L is a positive linear functional, i.e. $Lf \geq 0$ whenever $f \geq 0$. If $\{P_i\}_{i=1}^{+\infty}$ is an everywhere dense subset of D, then for sufficiently large $\mathcal{I}$, the set $X = \{P_i\}_{i=1}^{\mathcal{I}}$ is a Tchakaloff set, i.e.*

$$Lf = \sum_{j=1}^{\nu} w_j f(Q_j), \ \forall f \in \Phi \tag{1}$$

where $w_j > 0 \, \forall j$ and $\{Q_j\}_{j=1}^{\nu} \subset X \subset D$, with $\nu = card(\{Q_j\}) \leq N$.

Observe that we can apply this theorem to the case of algebraic cubature, setting $Lf = \int_D f(P) \; d\mu$ (being μ a positive measure on the compact set $D \subset \mathbb{R}^d$) and $\Phi = \mathbb{P}_n^d(D)$. We may term (1) in this case a "*Tchakaloff-like algebraic cubature rule*".

Furthermore, supposing that a Tchakaloff set X is available (how to construct one numerically for $d\mu = dxdy$ by a sequence of increasingly finer grids will be shown in Section 5), given any polynomial basis $\{\phi_j\}$ of $\mathbb{P}_n^d$, define the Vandermonde-like matrix

$$V = V_n(X) = (\phi_j(P_i))_{i,j} \in \mathbb{R}^{\mathcal{I} \times N} \ , \tag{2}$$

let $\gamma = \{\gamma_j\}$ be the vector of moments of the polynomial basis $\{\phi_j\}$ with respect to μ on D,

$$\gamma_j = \int_{\mathcal{S}} \phi_j(P) \, d\mu \ , \ \ j = 1, \dots, N \ , \tag{3}$$

and consider the (underdetermined) $N \times \mathcal{I}$ moment system

$$V^T u = \gamma \ . \tag{4}$$

Since X is a Tchakaloff set, there exists a sparse nonnegative solution u to the system above, whose nonvanishing components (i.e., the weights

$\{w_j\}$) are at most N and determine the corresponding reduced sampling points $\{Q_j\} \subset X$.

The computation of these nodes has been considered in several papers (see e.g. [21], [25]). To our knowledge, essentially two approaches have been developed to get these rules, i.e. via Linear Programming (LP) and Quadratic Programming (QP).

About the LP approach, it consists in solving via simplex-method

$$\begin{cases} \min_{u \geq 0} c^T u \\ V^T u = \gamma, \quad u \geq 0 \end{cases} \tag{5}$$

where the constraints identify a polytope (the feasible region) in $\mathbb{R}^M$ and the vector c is chosen to be linearly independent from the rows of V^T, so that the objective functional is not constant on the polytope [12], [25]. The solution is a vertex of the polytope, that has at least $M - N$ vanishing components, so that the nonzero components are the weights $\{w_j\}$ and determine the subset of nodes $\{Q_j\}$.

The QP based algorithm requires instead the solution of the NonNegative Least Squares (NNLS) problem

$$\text{compute } u^* : \|V^T u^* - \gamma\|_2 = \min_u \|V^T u - \gamma\|_2 \,,\; u \geq 0 \,, \tag{6}$$

in which u^* can be obtained by the well-known *Lawson-Hanson* active set optimization method [11], which seeks a sparse solution to (6), whose nonzero components are the weights $\{w_j\}$ and determine the subset of nodes $\{Q_j\}$. Its application gives a residual $\epsilon = \|V^T u^* - \gamma\|_2$ that is typically very small, say $< 10^{-14}$ for $n \leq 30$ in all our numerical tests.

Our numerical experience with the presently available Matlab software has shown that NNLS via Lawson-Hanson iterations usually performs better than LP in computing the weights $\{w_j\}$, at least for moderate degrees n [12]. On the other hand, in the recent paper [9] an acceleration of the Lawson-Hanson iterations has been discussed in the framework of compression of discrete probability measures and regression, that in perspective could be applied also to the cubature framework. Consequently all our codes are based on the application of Lawson-Hanson method.

We point out that there are several versions of NNLS codes available in Matlab. One is the built-in function `lsqnonneg`, while an open-source version present in the package NNLSlab in [18]. Other alternatives are often obtained by MEX files and in view of portability will not be used here.

Remark 4 In the cubature framework an algorithm termed *Recursive Halving Forest*, based on a hierarchical SVD, has been proposed in the doctoral dissertation [25]. Performances are reported for large scale problems (the size order of N, M is up to $10^3 - 10^4$), but essentially only in the compression of tensorial formulae on hypercubes. Unfortunately the software is not available and thus cannot be applied here as a comparison.

4 Implementing Tchakaloff-like cubature

Let $\mathcal{S}$ be Jordan domain whose boundary is defined by parametric splines $\tilde{x}(t)$, $\tilde{y}(t)$, as stated at the beginning of Section 2.

As in section 2, we suppose that $I_s^{(k)} = [t_s^{(k)}, t_{s+1}^{(k)}]$, $k = 1, \ldots, M$, $s = 1, \ldots, m_k - 1$, determines a partition of the interval $[a, b]$ and that $\partial\mathcal{S} = \{(\tilde{x}(t), \tilde{y}(t)),\ t \in [a, b]\}$, where the restriction of $\tilde{x}$, $\tilde{y}$ to $I_s^{(k)}$ is a polynomial of degree δ_k.

In this section we describe an algorithm that computes a Tchakaloff-like algebraic cubature formula of degree n over $\mathcal{S}$, i.e. such that

$$\int_{\mathcal{S}} p(x,y)\, dx\, dy = \sum_{j=1}^{\nu} w_j p(Q_j)\ ,\quad \nu \leq N = \dim(\mathbb{P}_n^2) = (n+1)(n+1)/2$$

for any bivariate polynomial $p \in \mathbb{P}_n^2$ (i.e., of total degree at most n).

We proceed as follows.

First we compute the *moments* γ_j of a certain polynomial basis $\{\phi_j\}$ of degree n over $\mathcal{S}$, i.e.

$$\gamma_j = \int_{\mathcal{S}} \phi_j(x,y)\, dx\, dy\ ,\ j = 1, \ldots, N\ .$$

These values can be efficiently obtained by means of Gauss-Green theorem and line integrals along the boundary, without the need of numerical cubature rules over $\mathcal{S}$, only using some Gauss-Legendre formulae.

In our Matlab software as polynomial basis $\{\phi_j\}$ we have used the (suitably ordered) total degree product Chebyshev basis $\{T_p(\alpha_1(x))T_q(\alpha_2(y))\}$, with $p + q \leq n$ and $(x, y) \in \mathcal{R}^* = [a_1, b_1] \times [a_2, b_2]$ (being $\mathcal{R}^*$ the smallest cartesian rectangle containing the domain, easily available when the all the monotone boxes are determined), where $T_h(\cdot) = \cos(h \arccos(\cdot))$ is the h-degree Chebyshev polynomial and $\alpha_i(s) = (2s - b_i - a_i)/(b_i - a_i)$, $s \in [a_i, b_i]$, $i = 1, 2$.

By Gauss-Green theorem (see e.g. [2]),

$$\gamma_j = \int_{\mathcal{S}} \phi_j(x,y)\, dx\, dy = \oint_{\partial\mathcal{S}} \Phi_j(x,y)\, dy \tag{7}$$

where, for some p, q,

$$\Phi_j(x,y) = \int \phi_j(x,y)\, dx = T_q(\alpha_2(y)) \int T_p(\alpha_1(x))\, dx$$

In particular, for $p = 0$

$$\int T_p(\alpha_1(x))\, dx = x,$$

for $p = 1$

$$\int T_p(\alpha_1(x))\, dx = \frac{b_1 - a_1}{4} \cdot \alpha_1^2(x),$$

while for $p \geq 2$

$$\int T_p(\alpha_1(x))\, dx = \frac{b_1 - a_1}{2} \cdot \left(\frac{p}{p^2-1} T_{p+1}(\alpha_1(x)) - \frac{x}{p-1} T_p(\alpha_1(x)) \right) .$$

Setting $P_{k,s} = (\tilde{x}(t_s^{(k)}), \tilde{y}(t_s^{(k)}))$ and denoting with $P_{k,s} \frown P_{k,s+1}$ the arc of $\partial\mathcal{S}$ joining $P_{k,s}$ with $P_{k,s+1}$, we have

$$\begin{aligned} \gamma_j = \oint_{\partial\mathcal{S}} \Phi_j(x,y)\, dy &= \sum_{k,s} \int_{P_{k,s} \frown P_{k,s+1}} \Phi_j(x,y)\, dy \\ &= \sum_{k,s} \int_{t_s^{(k)}}^{t_{s+1}^{(k)}} \Phi_j(\tilde{x}(t), \tilde{y}(t))\ \tilde{y}'(t) dt . \end{aligned} \tag{8}$$

Observing that

- since ϕ_j is a polynomial of total degree n then Φ_j is a polynomial of total degree $n+1$,
- in the interval $[t_s^{(k)}, t_{s+1}^{(k)}]$ both $\tilde{x}$, $\tilde{y}$ are polynomials of degree δ_k,

each integrand in the last sum of (8) is a polynomial of degree $(n+1)\delta_k + \delta_k - 1 = (n+2)\delta_k - 1$ that can be exactly integrated by a (shifted) Gauss-Legendre formula with $\lceil \frac{(n+2)\delta_k}{2} \rceil$ points. In such a way we can compute all the required moments.

We turn now to the construction of a Tchakaloff set in $\mathcal{S}$ from which we extract the nodes and positive weights of a Tchakaloff-like algebraic cubature rule. Setting $\mathcal{P}_0 = \emptyset$, at the ℓ-th iteration of the algorithm we define a set $\mathcal{P}_\ell$, consisting in the union of $\mathcal{P}_{\ell-1}$ with some points that belong to $\mathcal{S} \backslash \mathcal{P}_{\ell-1}$. To this purpose, we introduce a sequence of tensorial grids $\mathcal{M}_\ell$ in the rectangle $\mathcal{R}^* := [a_1, b_1] \times [a_2, b_2]$ containing $\mathcal{S}$, with $\mathcal{M}_\ell$ finer as

ℓ increases, determining by the *in-domain* algorithm, the new set $\mathcal{P}_\ell = \mathcal{P}_{\ell-1} \cup (\mathcal{M}_\ell \cap \mathcal{S})$.

Next, we apply the algorithm that computes the Tchakaloff formula with nodes $\{(x_i^{(\ell)}, y_i^{(\ell)})\}_{i=1,\dots,\nu_\ell}$ and positive weights $\{w_i^{(\ell)}\}_{i=1,\dots,\nu_\ell}$, $\nu_\ell \leq N$, finally testing if the so obtained rule is such that

$$\gamma_j^{(\ell)} = \sum_{i=1}^{\nu_\ell} w_i^{(\ell)} \phi_j(x_i^{(\ell)}, y_i^{(\ell)}),\ j = 1, \dots, N\ ,$$

well approximates the set of moments $\{\gamma_j\}$, i.e.

$$\|\gamma_j^{(\ell)} - \gamma_j\|_2 \leq \varepsilon \tag{9}$$

where ε is a tolerance fixed by the user. If (9) does not hold we iterate the procedure until (9) is satisfied or a maximum number of iterations is reached, providing in this case an error message.

In our implementation of Tchakaloff compression we used the Matlab built-in routine `lsqnonneg` but one can alternatively use the algorithm proposed in [18]. The fundamental fact that this procedure has finite termination comes from the result by Wilhelmsen [26] recalled by Theorem 2 above, since the set $\mathcal{P}_\ell$ becomes sufficiently dense in a finite number of iterations.

4.1 Boundary approximation and error estimates

As observed in [20], $\mathcal{S}$ can be an approximation of a certain compact Jordan domain $\Omega \subset \mathbb{R}^2$ with rectifiable boundary.

Assume that $\partial\Omega$ is a piecewise C^4 Jordan parametric curve $(x(t), y(t))$ with a finite number of breakpoints $\{V_k\}$ and $\partial\Omega \approx \partial\mathcal{S} := \cup_{k=1}^M V_k \frown V_{k+1}$, where each curvilinear side $V_k \frown V_{k+1}$ is tracked by a *cubic spline curve* (i.e. $\delta_k = 3$), interpolating an ordered subsequence of control points $P_{1,k} = V_k, P_{2,k}, \dots, P_{m_k-1,k}, P_{m_k,k} = V_{k+1}$. Moreover, suppose that it is adopted a *cumulative chordal* parametrization, i.e. each $V_k = (\tilde{x}(t^{(k)}), \tilde{y}(t^{(k)}))$ so that $I^{(k)} = [t^{(k)}, t^{(k+1)}]$, being $t^{(k+1)} = t^{(k)} + \sum_{j=1}^{m_k-1} h_{j,k}$, $h_{j,k} = |P_{j+1,k} - P_{j,k}|$. Then, by a result in [7], for $h = \max_{j,k} h_{j,k}$ sufficiently small the spline curve is simple (non self-intersecting), since not only $\partial\Omega$ but also its tangent vectors are well approximated and loops of $\partial\mathcal{S}$ are prevented. In such a way the spline interpolating curve is a Jordan curve, i.e. $\mathcal{S}$ is a Jordan domain itself as it is required by our construction.

On the other hand, denoting by $I_\Omega :\ I_\Omega(f) = \int_\Omega f(x,y)\,dxdy$ the integral functional on a compact domain Ω, by $I_n :\ I_n(f) = \sum_{j=1}^{\nu} w_j f(Q_j) \approx I_{\mathcal{S}}(f)$

the positive cubature formula functional (exact on polynomials in $\mathbb{P}_n^2$), by p_n^* the best polynomial in $\mathbb{P}_n^2$ of uniform approximation for $f \in C(\mathcal{S})$

$$\|f - p_n^*\|_{\infty,\mathcal{S}} = E_n(f,\mathcal{S}) = \inf_{p\in\mathbb{P}_n^2} \|f - p\|_{\infty,\mathcal{S}} \ ,$$

and setting $\Omega\Delta\mathcal{S} = (\Omega\backslash\mathcal{S})\cup(\mathcal{S}\backslash\Omega)$, with an approach similar to that adopted in [20], by a result in [16] on cubic spline approximation of planar curves we can write the chain of estimates (where ℓ denotes the *curve length* and $\mathcal{A}$ the *area measure*)

$$\begin{aligned}
|I_\Omega(f) - I_n(f)| \leq I_{\Omega\Delta\mathcal{S}}(|f|) + |I_\mathcal{S}(f) - I_n(f)| \leq \|f\|_{\infty,\Omega\Delta\mathcal{S}}\,\mathcal{A}(\Omega\Delta\mathcal{S})\\
+|I_\mathcal{S}(f) - I_\mathcal{S}(p_n^*)| + |I_\mathcal{S}(p_n^*) - I_n(p_n^*)| + |I_n(p_n^*) - I_n(f)|\\
\leq \|f\|_{\infty,\Omega\Delta\mathcal{S}}\,\mathcal{A}(\Omega\Delta\mathcal{S}) + (\|I_\mathcal{S}\| + \|I_n\|)\,\|f - p\|_{\infty,\mathcal{S}}\\
= \|f\|_{\infty,\Omega\Delta\mathcal{S}}\,\mathcal{A}(\Omega\Delta\mathcal{S}) + 2\,\mathcal{A}(\mathcal{S})\,E_n(f,\mathcal{S})\\
\leq \|f\|_{\infty,\Omega\Delta\mathcal{S}}\ \ell(\partial\Omega)\,e(h) + 2\,\mathcal{A}(\mathcal{S})\,E_n(f,\mathcal{S}) \ ,
\end{aligned}$$

and thus, since $e(h) = \mathcal{O}(h^4)$ and $\mathcal{S} \subset \Omega + B[0, e(h)]$ (the compact $e(h)$-neighborhood of Ω), we obtain the error estimate

$$|I_\Omega(f) - I_n(f)| = \mathcal{O}(h^4) + \mathcal{O}(E_n(f,\Omega_r)) \ , \tag{10}$$

provided that $f \in C(\Omega_r)$ with $\Omega_r = \Omega + B[0,r]$ suitable compact neighborhood of Ω and $e(h) \leq r$.

Now, the rate of $E_n(f,\Omega_r)$ can be estimated, as soon as we assume that Ω is a *Jackson compact set*, i.e. for every $s = 0,1,2,...$, there exists a positive integer α_s such that for $f \in C^{\alpha_s}(\Omega)$ there exists a constant $C_s(f)$ (depending on the partial derivatives of f up to order α_s) for which the Jackson-like inequality $E_n(f,\Omega) \leq C_s(f)\,n^{-s}$ holds for $n > s$; cf. [14] for a survey of known results on Jackson sets. Two basic examples are hypercubes with $\alpha_s = s+1$ and euclidean balls with $\alpha_s = s$. In particular a sufficient condition is that Ω be Whitney regular and admits a Markov polynomial inequality, which in turn holds whenever Ω is a finite union of real analytic images of hypercubes (a "subanalytic set"; cf. [13, 23] on deep subanalytic geometrical aspects of such a theory).

Indeed, in [14] it is proved that a finite union of Jackson sets is a Jackson set. Since $\Omega_r = \bigcup_{P\in\Omega} B[P,r] = \Omega \cup \bigcup_{P\in\partial\Omega} B[P,r]$, so that by compactness of $\partial\Omega$ it is the union of Ω with a finite number of euclidean balls of radius r centered at boundary points, and since a euclidean ball is a Jackson set with $\alpha_s = s$, we get from [14, Thm.1] that Ω_r is a Jackson set with $\alpha'_s = \max\{s, \alpha_s\}$, and thus

$$|I_\Omega(f) - I_n(f)| = \mathcal{O}(h^4) + \mathcal{O}(n^{-s}) \ , \quad n > s \ , \tag{11}$$

for f sufficiently regular on Ω_r.

Finally, one could observe that the estimates hold if the positive cubature formula is exact on polynomials in $\mathbb{P}_n^2$. On the other hand, our formula is only near-exact, in the sense that the moment residual is less than a tolerance ε by construction. In view of the estimates obtained in [21], which we do not report for brevity, the effect of ε is that the following overall error estimate holds

$$|I_\Omega(f) - I_n(f)| \le \|f\|_{\infty,\Omega\Delta\mathcal{S}} \ \ell(\partial\Omega)\, e(h) + \left(2\,\mathcal{A}(\mathcal{S}) + \sqrt{\mathcal{A}(\mathcal{S})}\right) \ E_n(f, \mathcal{S})$$

$$+\varepsilon\, \|f\|_{L^2(\mathcal{S})} = \mathcal{O}(h^4) + \mathcal{O}(E_n(f, \Omega_r)) + \mathcal{O}(\varepsilon) = \mathcal{O}(h^4) + \mathcal{O}(n^{-s}) + \mathcal{O}(\varepsilon) \ , \tag{12}$$

the latter again for f sufficiently regular on Ω_r and $n > s$.

5 Numerical experiments

The purpose of this section is twofold, on one side we test the new cubature algorithm and on the other the in-domain routines.

As already observed, the Matlab routine `splinegauss` proposed in [20] is not necessarily of PI-type on nonconvex spline curvilinear polygons $\mathcal{S}$, since in these instances some nodes could be outside the domain and some weights could be even negative. However, since it was based on a sequential use of the tensorial Gauss-Legendre formula via a panelization of the domain, the determination of the cubature rule was rather fast.

For relatively low degrees, say $n \le 10$, the new routine `splcub`, implementing the ideas developed in the previous sections, has similar cputimes but produces rules of *PI-type* with a number of nodes $\nu \le (n+1)(n+2)/2$. This fact has several implications:

- since all the weights positive, the rules are optimally stable, i.e. the cubature condition number

  $$\mathrm{cond}(\{w_i\}) = \frac{\sum_{i=1}^{\nu} |w_i|}{|\sum_{i=1}^{\nu} w_i|}$$

  is equal to 1, the best possible result for rules with $ADE \ge 0$;

- the number of nodes is particularly low (though usually not minimal);
- the rules are appropriate for problems were the sampling of the integrand is not possible outside the domain.

We implemented an algorithm, that can be sketched as follows:

1. define a mesh on the smallest rectangle $\mathcal{R}^* = [a_1, b_1] \times [a_2, b_2]$ containing the domain $\mathcal{S}$; in particular for $ADE = n$, then setting $\tau = \lfloor n^{1.5} \rfloor$, we considered the points $P_{ij} = (x_i, y_j)$ where
$$x_i = a_1 + i\frac{a_2 - a_1}{\tau - 1} \ , \ y_j = b_1 + j\frac{b_2 - b_1}{\tau - 1} \ , \ \ 0 \leq i, j \leq k - 1$$
i.e. a uniform tensor grid mesh $\mathcal{M}_1$, based on k equispaced points in each direction;

2. determine the points of $\mathcal{M}_1$ strictly inside the spline curvilinear polygon $\mathcal{S}$, say $\mathcal{P}_1$, by the *in-domain* algorithm developed in section 2;

3. compute the moments over $\mathcal{S}$ of the n-th total-degree product Chebyshev basis
$$\{T_p(\alpha_1(x))T_q(\alpha_2(y))\}, \ (x, y) \in [a_1, b_1] \times [a_2, b_2], \ 0 \leq p + q \leq n$$
with $\alpha_i(s) = (2s - b_i - a_i)/(b_i - a_i)$, $i = 1, 2$, by means of Gauss-Green theorem and Gauss-Legendre quadrature along the spline boundary arcs;

4. extract the nodes of a PI-type formula with $ADE = n$ from $\mathcal{P}_1$ and determine the relative weights using the discrete measure compression technique introduced in [21] and described in section 3 as Tchakaloff-like cubature.

In case of failure, i.e. the 2-norm of the moment error is bigger than a fixed tolerance, say $\varepsilon = 10^{-12}$, we proceed iteratively by defining finer and finer uniform tensor grids $\mathcal{M}_k$ (increasing the value of τ, as $\tau_{k+1} = \lfloor \beta \tau_k \rfloor$ with e.g. $\beta = 1.5$), determining at the k-th iteration those points belonging to $\mathcal{S}$, say $\mathcal{P}_k$, and performing step 4 of the algorithm above with the set $\bigcup_{i=1}^{k} \mathcal{P}_i$ instead of $\mathcal{P}_1$.

Possible ill-conditioning of Chebyshev-Vandermonde matrices arising at high ADE is managed by suitable discrete orthogonalization of the polynomial basis via the *economy size* QR factorization (as described for example in [21]). All the Matlab routines for the numerical experiments are available at [18] and have been tested on a PC with a 2.7 GHz Intel Core $i5$ CPU, with 16 GB of RAM.

We start our numerical tests by considering the region $\mathcal{S}_1$ containing $(0,0)$, whose boundary is defined componentwise by linear splines in the first four arcs and by cubic splines in the last five ones. The region is illustrated in Figure 4, on the left.

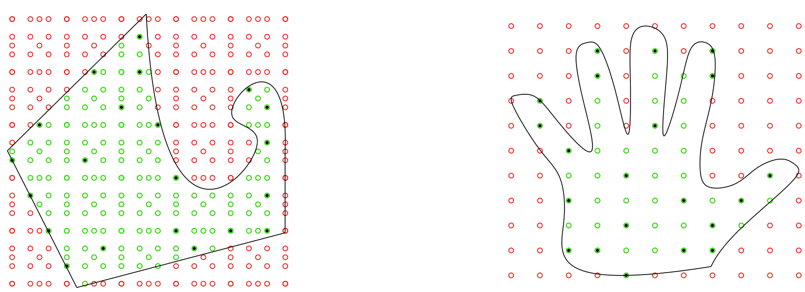

Figure 4: The spline curvilinear domains $\mathcal{S}_i$ with $i = 1, 2$, the grid points P outside the domain or on its boundary (in red), those inside the domain (in green) and the nodes of a cubature formula of PI-type for $n = 5$ (black dots).

In the present experiment, if the old routine `splinegauss` in [20] is applied to these tests, several nodes are external to the domain, and negative weights are present. The quality of the new rules is summarized in Table 1, which shows that they are numerically exact since the actual moment residuals are close to machine precision and with positive weights, being the cubature condition number equal to 1. For $ADE < 10$, the cputime is particularly low, making these formulas appealing for cubature within Virtual Elements Methods.

ADE	#	cond	moment res	cpu
5	21	1	$6.3e{-17}$	$0.01s$
10	66	1	$1.5e{-16}$	$0.05s$
15	136	1	$1.4e{-16}$	$0.09s$
20	231	1	$1.3e{-16}$	$0.81s$

Table 1: Cardinality # of the nodes, cubature conditioning and moment residual of the rule on domain $\mathcal{S}_1$, and cputime in seconds.

As second domain $\mathcal{S}_2$, we considered a hand-shaped domain containing $(0,0)$, whose boundary is defined by $\tilde{x}$, $\tilde{y}$, that are both cubic splines satis-

ADE	#	cond	moment res	cpu
5	21	1	$5.1e{-}18$	$0.01s$
10	66	1	$3.6e{-}18$	$0.03s$
15	136	1	$3.2e{-}18$	$0.09s$
20	231	1	$4.5e{-}18$	$0.62s$

Table 2: As in table 2 for the hand-shaped domain $\mathcal{S}_2$.

fying *not-a-knot* conditions, having 37 control points (see Figure 4, on the right). The numerical results are reported in Table 2 and are essentially similar to those on the domain $\mathcal{S}_1$, confirming the good quality of the rules.

Inn Figure 5 we illustrate the performance of these new cubature rules to determine

$$I = \int_{\mathcal{S}_i} (c_0 + c_1 x + c_2 y)^n \, dx \, dy \;, \quad i = 1, 2 \;,$$

making 100 trials with uniform random coefficients $c_j \in (0,1)$, $j = 0,1,2$ and $ADE = n$, $n = 5, 10, 15, 20$. The reference values of I have been determined by applying Gauss-Green theorem and Gaussian quadrature along the spline boundary, taking into account that an x-primitive of the polynomial can be readily computed. We have plotted with a dot the relative error RE_k made by the rule (log scale), and by a larger circle the logarithmic average on all the trials, i.e. $\sum_{k=1}^{100} \log(RE_k)/100$.

The tests show that in spite of the fact that the moments are computed close to machine precision, there is a little deterioration of the logarithmic average error for $n = 20$, while for lower degrees this value remains between -15 and -14. Notice that an ADE greater than 10 is usually beyond what is needed for example in VEM applications.

As a further illustration, we report in Tables 3-4 the relative errors made by the Tchakaloff-like rules when approximating $\int_{\mathcal{S}_i} f_k(x,y)\, dx\, dy$, where

$$\begin{aligned} f_1(x,y) &= \exp(-(x^2+y^2)) \;, \\ f_2(x,y) &= (x^2+y^2)^{11/2} \;, \\ f_3(x,y) &= (x^2+y^2)^{1/2} \;, \end{aligned} \tag{13}$$

that are examples of functions with different degree of regularity on each domain $\mathcal{S}_i$, $i = 1, 2$. The reference values of the integrals are those obtained by `splinegauss` in [20] with $ADE = 100$. As expected, in both the domains

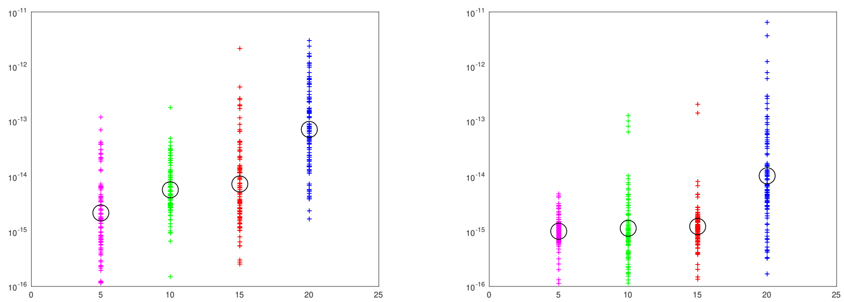

Figure 5: *Dots*: Relative errors RE_k, $k = 1, \dots, 100$, on cubature over random polynomials $(c_0 + c_1 x + c_2 y)^n$ on $\mathcal{S}_1$ (on the left) and $\mathcal{S}_2$ (on the right). *Circles*: average logarithmic error, i.e. $10^{\sum_{k=1}^{100} \log(RE_k)/100}$. The abscissae are the ADE of the formula and are equal to 5, 10, 15, 20.

the quality of the approximation worsens for less regular integrands (indeed $f_1 \in C^\infty(\mathcal{S}_i)$, whereas $(0,0) \in \mathcal{S}_i$ is a singular point for the first derivatives of f_3 and for 11-th derivatives of f_2).

ADE	f_1	f_2	f_3
5	1.5e−03	2.0e−01	1.5e−02
10	2.8e−08	4.4e−05	1.4e−03
15	4.3e−11	2.5e−07	1.7e−04
20	1.0e−15	2.0e−09	2.2e−05

Table 3: Relative errors of the new rules on the domain $\mathcal{S}_1$, with $ADE = 5, 10, \dots, 20$.

ADE	f_1	f_2	f_3
5	1.9e−08	1.4e−01	1.2e−02
10	2.0e−15	1.1e−04	1.1e−03
15	1.1e−15	6.0e−08	1.2e−03
20	1.5e−15	1.8e−09	5.0e−04

Table 4: As in Table 3 on the hand-shaped domain $\mathcal{S}_2$.

As additional information, we show in Table 5 the cputimes necessary to process K in-domain operations, on the regions $\mathcal{S}_i$, $i = 1, 2$. To this purpose we have written the Matlab routine `incurvpolygon`, that implements the ideas introduced in section 2.

K	$\mathcal{S}^{(1)}$	$\mathcal{S}^{(2)}$
10^2	$4e{-}03s$	$7e{-}03s$
10^3	$5e{-}03s$	$1e{-}02s$
10^4	$3e{-}02s$	$6e{-}02s$
10^5	$2e{-}01s$	$3e{-}01s$
10^6	$3e{+}00s$	$3e{+}00s$

Table 5: Cputime of the application of the *in-domain* algorithm for K points of the surrounding rectangles, respectively to $\mathcal{S}^{(1)}$ and $\mathcal{S}^{(2)}$ of Figure 4.

The results are quite satisfactory concerning the application of the cubature algorithm at mild degrees. A comparison with the cputime needed to determine the rules shows that at high degrees the bottleneck is actually the application of the current implementation of Lawson-Hanson algorithm.

Remark 5 (Nested rules). We point out that, if it is available a formula of PI-type with $ADE = n$, then one can extract those with $ADE < n$, i.e. *nested rules*, by successive applications of Tchakaloff-like cubature compression (as described in [22]).

References

[1] F. Aldakheel, B. Hudobivnik, E. Artioli, L. Beirão da Veiga and P. Wriggers, Curvilinear Virtual Elements for Contact Mechanics, ResearchGate preprint (2020), `https://www.researchgate.net/publication/340898242`.

[2] T.M. Apostol, Calculus, 2nd ed., vol. II, Blaisdell, 1969.

[3] E. Artioli, L. Beirão da Veiga and 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] B. Bauman, A. Sommariva and M. Vianello, Compressed cubature over polygons with applications to optical design, J. Comput. Appl. Math. 370 (2020).

[6] L. Beirão da Veiga, A. Russo and G. Vacca, The Virtual Element Method with curved edges, ESAIM Mathematical Modelling and Numerical Analysis 53 (2019), 375–404.

[7] L. Bos and M. Vianello, On simple approximations to simple curves, Dolomites Res. Notes Approx. DRNA 3 (2010), 1–6.

[8] P.J. Davis, A construction of nonnegative approximate quadratures, Math. Comp. 21 (1967), 578–582.

[9] M. Dessole, F. Marcuzzi and M. Vianello, Accelerating the Lawson-Hanson NNLS solver for large-scale Tchakaloff regression designs, Dolomites Res. Notes Approx. DRNA 13 (2020), 20–29.

[10] K. Hormann and A. Agathos, The point in polygon problem for arbitrary polygons, Comput. Geom. Theory Appl. 20, n.3 (2001), 131–144.

[11] C.L. Lawson and R.J. Hanson, Solving least squares problems, Classics in Applied Mathematics 15, SIAM, Philadelphia, 1995.

[12] F. Piazzon, A. Sommariva and M. Vianello, Caratheodory-Tchakaloff Subsampling, Dolomites Res. Notes Approx. DRNA 10 (2017), 5–14.

[13] W. Pleśniak, Multivariate polynomial inequalities via pluripotential theory and subanalytic geometry methods, Banach Center Publ. 72 (2006), 251–261.

[14] W. Pleśniak, Multivariate Jackson Inequality, J. Comput. Appl. Math. 233 (2009), 815–820.

[15] M. Putinar, A note on Tchakaloff theorem, Proc. Amer. Math. Soc. 125 (1997), 2409–2414.

[16] M. Sakai and R.A. Usmani, On orders of approximation of plane curves by parametric cubic splines, BIT Numerical Mathematics 30 (1990), 735–741.

[17] G. Santin, A. Sommariva and M. Vianello, An algebraic cubature formula on curvilinear polygons, Appl. Math. Comput. 217 (2011), 10003–10015.

[18] M. Slawski, Non-negative least squares: comparison of algorithms, `https://sites.google.com/site/slawskimartin/code`.

[19] A. Sommariva, Quadrature and Cubature Codes, available at: `https://www.math.unipd.it/~alvise/software.html`.

[20] A. Sommariva and M. Vianello, Gauss-Green cubature and moment computation over arbitrary geometries, J. Comput. Appl. Math., 231 (2009), 886–896.

[21] A. Sommariva and M. Vianello, Compression of multivariate discrete measures and applications, Numer. Funct. Anal. Optim. 36 (2015), 1198–1223.

[22] A. Sommariva and M. Vianello, Nearly optimal nested sensors location for polynomial regression on complex geometries, Sampl. Theory Signal Image Process. 17 (2018), 95–101.

[23] J. Stasica, The Whitney condition for subanalytic sets, Zeszyty Nauk. Uniw. Jagiellon. Prace Mat. 23 (1982), 211–221.

[24] V. Tchakaloff, Formules de cubatures mécaniques à coefficients non négatifs, (French) Bull. Sci. Math. 81 (1957), 123–134.

[25] M. Tchernychova, Caratheodory cubature measures, Ph.D. dissertation in Mathematics (supervisor: T. Lyons), University of Oxford, 2015.

[26] D.R. Wilhelmsen, A Nearest Point Algorithm for Convex Polyhedral Cones and Applications to Positive Linear approximation, Math. Comp. 30 (1976), 48–57.

[27] Q. Zhang, Highly Accurate Lagrangian Flux Calculation via Algebraic Quadratures on Spline-Approximated Donating Regions, Comput. Methods Appl. Mech. Eng. 264 (2013), 191–204.

[28] Q. Zhang, On donating regions: Lagrangian flux through a fixed curve, SIAM Rev. 55 (2013), 443–461.