

TetraFreeQ: tetrahedra-free quadrature on polyhedral elements

Alvise Sommariva^{a,b}, Marco Vianello^{a,b}

^aUniversity of Padova, Italy

^bMember of the INdAM Research group GNCS

Abstract

In this paper we provide a tetrahedra-free algorithm to compute low-cardinality quadrature rules with a given degree of polynomial exactness, positive weights and interior nodes on a polyhedral element with arbitrary shape. The key tools are the notion of Tchakaloff discretization set and the solution of moment-matching equations by Lawson-Hanson iterations for NonNegative Least-Squares. Several numerical tests are presented. The method is implemented in Matlab as open-source software.

Keywords: Polyhedral elements, tetrahedra-free algebraic quadrature, Positive Interior rules, Tchakaloff theorem, Tchakaloff sets, Davis-Wilhelmsen theorem, NonNegative Least Squares.

2020 MSC: 65D32.

1. Introduction

Let $\Omega \subset \mathbb{R}^3$ is a polyhedron (either convex or nonconvex or even multiply-connected), and suppose that for a given continuous function $f \in C(\Omega)$ one needs to approximate its integral over Ω by a quadrature rule with nodes $\{Q_j\}_{j=1,\dots,\nu} \subset \Omega$ where $Q_j = (x_j, y_j, z_j)$ and positive weights $\{w_j\}_{j=1,\dots,\nu}$,

$$I_\Omega(f) = \int_\Omega f(x, y, z) \, dx \, dy \, dz \approx I_n(f) = \sum_{j=1}^{\nu} w_j f(Q_j), \quad (1)$$

that has Algebraic Degree of Exactness $ADE = n$, i.e. $I_\Omega(p) = I_n(p)$ for all $p \in \mathbb{P}_n^3$ (the space of trivariate polynomials of total degree not exceeding n). Such a kind of rules are often called PI (Positive Interior) in the literature and are optimally stable by the positivity of the weights, being the quadrature conditioning $\sum_j |w_j| / \left| \sum_j w_j \right| = 1$. As a relevant motivation, we may recall that the algebraic quadrature problem on polyhedra has been extensively studied in the literature, especially during the last decade, being at the core of many discretization methods for PDEs that use polyhedral meshes, such as polyhedral FEM and discontinuous Galerkin, as well as the more recent VEM. We may quote among others, without any pretence of exhaustivity, the papers [1, 4, 15, 23, 24] with the numerous references therein.

Despite of such manifest interest in the framework of numerical PDEs, open-source numerical codes to compute PI algebraic quadrature rules on polyhedra, in particular Matlab codes, do not seem to be readily available. On the other hand, differently from other approaches that do not provide an algebraic quadrature formula but a numerical integration algorithm, the availability of low-cardinality quadrature formulas allows to compute efficiently integrals involving forcing terms, without the need of an explicit polynomial approximation of such terms (cf. [1, Rem. 6]). Moreover, since the polyhedral quadrature formulas can be computed once and for all for a fixed polyhedral mesh, even more flexibility is gained when different forcing terms have to be tested, or more generally when parametrized PDEs have to be solved, for example by reduced basis methods or other model order reduction techniques (cf., e.g., [10]).

*Corresponding author

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

Indeed, the main purpose of the present work is to provide an algorithm implemented by an open-source Matlab package, named TETRAFREEQ, for the computation of tetrahedra-free PI algebraic quadrature rules on arbitrary polyhedra (either convex or nonconvex or even multiply-connected), with cardinality $\nu \leq N$ where

$$N = N_n = \dim(\mathbb{P}_n^3) = (n+1)(n+2)(n+3)/6. \quad (2)$$

Existence and computability of such rules are guaranteed by the well-known Tchakaloff theorem [25] and the less known Wilhelmsen theorem on ‘‘Tchakaloff sets’’ [28], the latter being at the core of our approach as described in the next section.

2. Tetrahedra-free quadrature

In this section we show how to determine a quadrature rule with $ADE = n$, positive weights and interior nodes, on a general polyhedron Ω , without needing a decomposition of the domain by means of nonoverlapping tetrahedra (tetrahedra-free quadrature, for short). It should be recalled that avoiding ‘‘sub-tessellations’’ for integration on polyhedral elements is a common requirement in the numerical PDEs framework, especially with high-order methods.

In the sequel the polyhedron is assumed to be given via its polygonal faces, say $\{\mathcal{F}_i\}$, each with counterclockwise oriented vertices w.r.t. the outward normal. The procedure works essentially as follows:

Algorithm TetraFreeQ (tetrahedra-free quadrature on a general polyhedron Ω)

(i) compute the moment array

$$\gamma = \{\gamma_k\} = \left\{ \int_{\Omega} \phi_k(x, y, z) dx dy dz \right\} \quad (3)$$

of a certain polynomial basis $\{\phi_k\}$ of \mathbb{P}_n^3 , $k = 1, \dots, N$, by the divergence theorem and bivariate quadrature rules with ADE equal to $n+1$ on the polyhedron faces $\{\mathcal{F}_i\}_{i=1, \dots, M}$

(ii) using an *in-polyhedron* routine [11], in view of Wilhelmsen theorem [28], determine iteratively a sufficiently dense pointset $\{P_l\}_{l=1, \dots, L}$ inside Ω (a so-called ‘‘Tchakaloff set’’) such that the underdetermined moment-matching system

$$V^t u = \gamma, \quad \text{where } V = [v_{lk}] = [\phi_k(P_l)] \in \mathbb{R}^{L \times N} \quad (4)$$

has a nonnegative solution u with at most $N \leq L$ positive components, computed via Lawson-Hanson NNLS algorithm [13] applied to

$$\min_{u \geq 0} \|V^t u - \gamma\|_2 \quad (5)$$

(iii) once $\{P_l\}_{l=1, \dots, L}$ and the weight vector u are determined, the nonzeros of u finally select the nodes of a PI-type quadrature rule with cardinality at most N .

In spite of the simplicity of this approach, there are many theoretical and practical aspects that deserve explanations. We stress that the key tools are Wilhelmsen theorem on the existence of ‘‘Tchakaloff sets’’ within sufficiently dense discrete subsets (a relevant but somehow overlooked result of multivariate quadrature theory), and the capability of Lawson-Hanson active-set iterative algorithm to compute a sparse solution to the NonNegative Least-Squares problem (made more efficient by the recent Deviation Maximization approach to QR factorization [6, 7]).

2.1. Moment computation

For an implementation of the algorithm sketched above, we intend to compute first the moments on the polyhedron Ω of the product Chebyshev basis of total degree n , relative to the smallest parallelepiped $\mathcal{R} = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ containing Ω .

In other words, defined the scaled Chebyshev polynomial in $[a, b]$

$$\tilde{T}_m^{(a,b)}(t) := T_m \left(\frac{2}{b-a} \cdot \left(t - \frac{a+b}{2} \right) \right) \quad (6)$$

where T_m is the Chebyshev polynomial of first kind, of degree m , we compute the value of the moments

$$\gamma_\alpha = \int_{\Omega} \tilde{T}_{\alpha_1}^{(a_1,b_1)}(x) \tilde{T}_{\alpha_2}^{(a_2,b_2)}(y) \tilde{T}_{\alpha_3}^{(a_3,b_3)}(z) \, dx \, dy \, dz$$

for all the triples $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ with $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{N}$ and $0 \leq \alpha_1 + \alpha_2 + \alpha_3 \leq n$.

To this purpose, we recall that in view of the divergence theorem applied to a continuously differentiable vector field $\mathbf{F} = (f, g, h)$ defined on a neighborhood of Ω , we have

$$\int_{\Omega} \nabla \cdot \mathbf{F} \, dx \, dy \, dz = \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} \, dS \quad (7)$$

where $\mathbf{n} = (n_1, n_2, n_3)$ is the outward pointing unit normal at each point on the boundary $\partial\Omega$ and $\nabla \cdot \mathbf{F} = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z}$ is the divergence of \mathbf{F} . As a particular case of (7), we consider $\mathbf{F} = (f, 0, 0)$ obtaining

$$\int_{\Omega} \frac{\partial f(x, y, z)}{\partial x} \, dx \, dy \, dz = \int_{\partial\Omega} n_1(x, y, z) f(x, y, z) \, dS, \quad (8)$$

and we apply (8) to a certain f such that

$$\frac{\partial f(x, y, z)}{\partial x} = \tilde{T}_{\alpha_1}^{(a_1,b_1)}(x) \tilde{T}_{\alpha_2}^{(a_2,b_2)}(y) \tilde{T}_{\alpha_3}^{(a_3,b_3)}(z).$$

In this regard, we exploit a fundamental property of Chebyshev polynomials of first kind, namely that for $m \geq 2$

$$\int_0^x T_m(t) dt = \frac{T_{m+1}(x)}{2(m+1)} - \frac{T_{m-1}(x)}{2(m-1)}. \quad (9)$$

Consequently, setting $s = \frac{2}{b-a} \cdot \left(t - \frac{a+b}{2} \right)$, by (6) and (9), for $m \geq 2$,

$$\begin{aligned} \int_0^x \tilde{T}_m^{(a,b)}(t) dt &= \int_0^x T_m \left(\frac{2}{b-a} \cdot \left(t - \frac{a+b}{2} \right) \right) dt \\ &= \int_{-\frac{a+b}{b-a}}^{\frac{2}{b-a}(x - \frac{a+b}{2})} T_m(s) \frac{2}{b-a} ds \\ &= \frac{2}{b-a} \left(\int_{-\frac{a+b}{b-a}}^0 T_m(s) ds + \int_0^{\frac{2}{b-a}(x - \frac{a+b}{2})} T_m(s) ds \right) \\ &= \frac{2}{b-a} \left(\int_{-\frac{a+b}{b-a}}^0 T_m(s) ds + \frac{T_{m+1}(\frac{2}{b-a}(x - \frac{a+b}{2}))}{2(m+1)} - \frac{T_{m-1}(\frac{2}{b-a}(x - \frac{a+b}{2}))}{2(m-1)} \right) \\ &= \frac{2}{b-a} \left(\int_{-\frac{a+b}{b-a}}^0 T_m(s) ds + \frac{\tilde{T}_{m+1}^{(a,b)}(x)}{2(m+1)} - \frac{\tilde{T}_{m-1}^{(a,b)}(x)}{2(m-1)} \right). \end{aligned} \quad (10)$$

Thus, being the term $\frac{2}{b-a} \int_{-\frac{a+b}{b-a}}^0 T_m(s) ds$ constant, setting

$$\mathcal{Z}_m^{(a,b)}(x) = \frac{1}{b-a} \left(\frac{\tilde{T}_{m+1}^{(a,b)}(x)}{m+1} - \frac{\tilde{T}_{m-1}^{(a,b)}(x)}{m-1} \right)$$

we get that for $m \geq 2$

$$\tilde{T}_m^{(a,b)}(x) = \frac{\partial}{\partial x} \mathcal{Z}_m^{(a,b)}(x). \quad (11)$$

It can be easily checked that (11) holds also in the case $m = 0$ and $m = 1$, by defining respectively

$$\mathcal{Z}_0^{(a,b)}(x) = x - \frac{a+b}{2}, \quad \mathcal{Z}_1^{(a,b)}(x) = \frac{1}{b-a} \left(x - \frac{a+b}{2} \right)^2,$$

and that $\mathcal{Z}_m^{(a,b)} \in \mathbb{P}_{m+1}$.

Now, denoting by $\{\mathcal{F}_i\}_{i=1,\dots,M}$ the polygonal faces that compose the boundary of the polyhedron Ω , where $\mathbf{n}^{(i)} = (n_1^{(i)}, n_2^{(i)}, n_3^{(i)})$ is the outward normal of \mathcal{F}_i , $i = 1, \dots, M$, by the form of the divergence theorem (8) for

$$f_\alpha(x, y, z) = \mathcal{Z}_{\alpha_1}^{(a_1, b_1)}(x) \tilde{T}_{\alpha_2}^{(a_2, b_2)}(y) \tilde{T}_{\alpha_3}^{(a_3, b_3)}(z)$$

from the application of (11) with $a = a_1$ and $b = b_1$, we have

$$\begin{aligned} \gamma_\alpha &= \int_{\Omega} \tilde{T}_{\alpha_1}^{(a_1, b_1)}(x) \tilde{T}_{\alpha_2}^{(a_2, b_2)}(y) \tilde{T}_{\alpha_3}^{(a_3, b_3)}(z) \, dx \, dy \, dz \\ &= \int_{\Omega} \frac{\partial}{\partial x} \mathcal{Z}_{\alpha_1}^{(a_1, b_1)}(x) \tilde{T}_{\alpha_2}^{(a_2, b_2)}(y) \tilde{T}_{\alpha_3}^{(a_3, b_3)}(z) \, dx \, dy \, dz \\ &= \int_{\Omega} \frac{\partial}{\partial x} \left(\mathcal{Z}_{\alpha_1}^{(a_1, b_1)}(x) \tilde{T}_{\alpha_2}^{(a_2, b_2)}(y) \tilde{T}_{\alpha_3}^{(a_3, b_3)}(z) \right) \, dx \, dy \, dz \\ &= \int_{\partial\Omega} n_1(x, y, z) \mathcal{Z}_{\alpha_1}^{(a_1, b_1)}(x) \tilde{T}_{\alpha_2}^{(a_2, b_2)}(y) \tilde{T}_{\alpha_3}^{(a_3, b_3)}(z) \, dS \\ &= \sum_{i=1}^M \int_{\mathcal{F}_i} n_1^{(i)} \mathcal{Z}_{\alpha_1}^{(a_1, b_1)}(x) \tilde{T}_{\alpha_2}^{(a_2, b_2)}(y) \tilde{T}_{\alpha_3}^{(a_3, b_3)}(z) \, dS. \end{aligned} \quad (12)$$

Since the outward normal n_1 is constant on the face \mathcal{F}_i , so is $n_1^{(i)}$, and thus the integrand

$$n_1^{(i)} \mathcal{Z}_{\alpha_1}^{(a_1, b_1)}(x) \tilde{T}_{\alpha_2}^{(a_2, b_2)}(y) \tilde{T}_{\alpha_3}^{(a_3, b_3)}(z)$$

is a polynomial of degree not exceeding $n + 1$ on the i -th face, with $i = 1, \dots, M$, and consequently each moment γ_α can be recovered by a quadrature rule with ADE equal to $n + 1$ for all the polygons $\mathcal{F}_1, \dots, \mathcal{F}_M$.

Having this in mind, let $\mathcal{F} \subset \partial\Omega$ be a (polygonal) face of the polyhedron Ω , and suppose that the vertices are oriented counterclockwise w.r.t. its outward normal \mathbf{n} . Since it is easier to determine a quadrature rule on a planar polygon \mathcal{F}^* , we apply a rototranslation Φ , such that $\mathcal{F}^* = \Phi(\mathcal{F})$ is a polygon belonging to the plane $\pi^* = \{(x, y, 0), x, y \in \mathbb{R}\}$, oriented counterclockwise w.r.t. the vector $(0, 0, 1)$.

A quadrature formula on the planar polygon \mathcal{F}^* with internal nodes $\{P_i\}$ and positive weights $\{w_i\}$, having ADE equal to $n + 1$, can be determined as described in [3], by applying a rule with ADE equal to $n + 1$ on each element of a minimal triangulation of \mathcal{F}^* or alternatively by a triangulation free routine as discussed in [21]. Since the rototranslation Φ is an affine (bijective) map with Jacobian modulus equal to 1, we conclude that $\{\Phi^{-1}(P_i)\}$ and $\{w_i\}$ are the nodes and the weights of a quadrature rule over the polygonal face $\mathcal{F} \subset \partial\Omega$, with ADE equal to $n + 1$ and consequently we are able to compute the set of moments $\{\gamma_\alpha\}$.

Remark 2.1. The mapping ϕ can be easily obtained by applying a rotation R to the plane π containing the vertices of the polygon \mathcal{F} , so that $R\pi$ is parallel to the xy -plane $\pi^* = \{(x, y, 0), x, y \in \mathbb{R}\}$ and then by applying a vertical translation τ so that $\hat{\mathcal{F}} := R\mathcal{F} + \tau \subset \pi^*$. If a counterclockwise orientation of \mathcal{F} is given by the user, then it is straightforward to obtain the same for $\hat{\mathcal{F}}$. For the quadrature routines over planar polygons it is a typical requirement that the orientation is counterclockwise. If the planar polygon $\hat{\mathcal{F}}$

is oriented clockwise w.r.t. the z -axis then, applying an additional rotation, we can take as \mathcal{F}^* the polygon symmetric to $\hat{\mathcal{F}}$ w.r.t. the x -axis, otherwise it is sufficient to set $\mathcal{F}^* = \hat{\mathcal{F}}$. We recall that a necessary and sufficient condition for a polygon with vertices $(x_j, y_j, 0)$, $j = 1, \dots, m$ to be oriented counterclockwise is that its signed area $\sum_{j=1}^{m-1} (x_{j+1} - x_j)(y_{j+1} + y_j)$ be negative (for further details see [27]).

In some instances, instead of the orientation of the vertices of \mathcal{F} , it is available its outward normal \mathbf{n} (w.r.t. Ω). In this case, the procedure is the similar to that of the previous remark. If in particular, after the application of the rotation R (and the translation τ), the normal to $\hat{\mathcal{F}}$ correspond to $\Phi(\mathbf{n}) = (0, 0, -1)$ then as \mathcal{F}^* we can consider the polygon symmetric to $\hat{\mathcal{F}}$ w.r.t. the x -axis, otherwise we set $\mathcal{F}^* = \hat{\mathcal{F}}$.

2.2. Tchakaloff sets and computation of Tchakaloff-like rules

In the previous subsection we have implemented step (i) of the Algorithm TetraFreeQ sketched at the beginning of Section 2. With a little abuse of notation, we shall consider an ordering of the multi-indices $\alpha = (\alpha_1, \alpha_2, \alpha_3)$, $0 \leq \alpha_1 + \alpha_2 + \alpha_3 \leq n$, e.g. the graded lexicographical ordering, and we shall term $\{\phi_k(x, y, z)\}$ the product Chebyshev basis and $\{\gamma_k\}$ the corresponding moments, $1 \leq k \leq N$.

Now, step (ii) of Algorithm TetraFreeQ rests on the following relevant (though often overlooked) result proved by Wilhelmsen in [28] where the key notion of *Tchakaloff set* is introduced (extending a result of Davis [8] to rather general functional spaces, including polynomial spaces).

Theorem 2.1. (*Wilhelmsen, 1976*) *Let Ψ be the linear span of continuous, real-valued, linearly independent functions $\{\phi_k\}_{k=1, \dots, N}$ defined on a compact set $\Omega \subset \mathbb{R}^d$. Assume that Ψ satisfies the Krein condition (i.e. there is at least one $f \in \Psi$ which does not vanish on Ω) and that L is a positive linear functional, i.e. $Lf > 0$ for every $f \geq 0$ not vanishing everywhere in Ω .*

If $\{P_l\}_{l=1}^\infty$ is an everywhere dense subset of Ω , then for sufficiently large L , the set $X = \{P_l\}_{l=1, \dots, L}$ is a Tchakaloff set, i.e. there exist weights $w_j > 0$, $j = 1, \dots, \nu$, and nodes $\{Q_j\}_{j=1, \dots, \nu} \subset X \subset \Omega$, with $\nu = \text{card}(\{Q_j\}) \leq N$, such that

$$Lf = \sum_{j=1}^{\nu} w_j f(Q_j), \quad \forall f \in \Psi. \quad (13)$$

In view of this existence result, we can the implement step (ii) in our framework, that is $Lf = I_\Omega(f)$ and $\Psi = \mathbb{P}_n^3$, by the following iterative substeps:

- (ii1) generate a set of uniform random or low-discrepancy points (e.g. Halton points) with cardinality K in the smallest box $[a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ containing Ω
- (ii2) determine those points belonging to Ω by an in-polyhedron routine like [11], say $X = \{P_l\}_{l=1, \dots, L}$ with $L \leq K$, and compute the Vandermonde-like matrix $V = [v_{lk}] = [\phi_k(P_l)] \in \mathbb{R}^{L \times N}$
- (ii3) solve the NNLS problem $\min_{u \geq 0} \|V^t u - \gamma\|_2$ by Lawson-Hanson active-set method
- (ii4) if the residual $\|V^t u - \gamma\|_2 < \varepsilon$ (where ε is a given tolerance)
 - then set $\{w_j\} = \{u_l > 0\}$ and $\{Q_j\} = \{P_l : u_l > 0\}$
 - else goto (ii1) increasing K , e.g. by a fixed factor $\theta > 1$

We stress that since the set of random points becomes *denser* along the iterations, in view of Wilhelmsen theorem above termination of the iterative cycle is (at least theoretically) guaranteed. Substep (ii3) is based on the fact that Lawson-Hanson algorithm computes a sparse solution of the NNLS problem, and can be performed in Matlab by resorting to the built-in routine `lsqnonneg`. On the other hand, a substantial acceleration can be obtained via the recent Deviation-Maximization approach to QR factorizations which are the core of Lawson-Hanson method; cf. [5, 6, 7] for a detailed analysis of this approach.

Remark 2.2. (Tetrahedra-based quadrature)

As already stressed, the aim of the present work is to provide a completely “tetrahedra-free” approach to quadrature on general polyhedral elements, implemented in Matlab. On the other hand, a classical approach for polyhedral quadrature, whenever a nonoverlapping “tetrahedralization” of the polyhedron is available, consists in collecting and summing up the PI-formulas on the single tetrahedra by integration additivity.

A first problem consists in the computation of the tetrahedralization. If Ω is convex or star-shaped with a known center, this operation is straightforward, but the task becomes not trivial for a general polyhedron that may have a difficult geometry. We recall, incidentally, that differently from the 2D case of polygons, there exist polyhedra that cannot be triangulated using only their vertices (e.g. the well-known Schönardt polyhedron, cf. [17]).

Moreover, though Matlab provides a *minimal* triangulation of a general polygonal region via the environment `polyshape`, similar codes do not seem available for polyhedra. As an alternative, sometimes useful, the built-in `alphashape` command creates a bounding volume that envelops a given 3D point cloud pointset. If the polyhedron of interest can be defined in such a way, then `alphaTriangulation` returns a tetrahedralization that defines the domain of the alpha shape (see [14] for additional details).

A second problem concerns the rule to be used in each tetrahedron. For degrees of exactness up to $n = 20$, there are in literature several PI-type rules on the reference tetrahedron (the simplex) with vertices $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 0)$, $(0, 0, 1)$, which have near-minimal cardinality (see e.g. [12, 18] and Table 1). For $n > 20$, one can use the well-established Stroud rule [22, p.28-32], again of PI-type, that in general is not minimal but still easy to be implemented and with a moderate cardinality equal to $\lceil \frac{n+1}{2} \rceil^3$. All these rules are typically written in barycentric coordinates $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ and can be applied to a general tetrahedron with vertices A, B, C, D by the standard change of coordinates $\lambda_1 A + \lambda_2 B + \lambda_3 C + \lambda_4 D$.

deg	card	deg	card	deg	card	deg	card
1	1	6	23	11	94	16	247
2	4	7	31	12	117	17	288
3	6	8	44	13	144	18	338
4	11	9	57	14	175	19	390
5	14	10	74	15	207	20	448

Table 1: Cardinality of near-minimal rules on the reference tetrahedron.

Of course the overall cardinality of the resulting rule is proportional to the number of tetrahedra and can be much larger than $N = (n + 1)(n + 2)(n + 3)/6 = \dim(\mathbb{P}_n^3)$. In these cases, one can resort to Tchakaloff-like compression of the corresponding discrete measure, obtaining a rule with at most N re-weighted nodes that are a subset of the starting ones. Such a compression consists in computing a sparse positive solution of a moment-matching system like (4), where the points and moments are given by the high-cardinality rule itself. This approach inserts in the more general setting of “Caratheodory-Tchakaloff discrete measure compression”, and can be implemented by both Linear and Quadratic Programming methods; cf. e.g. [9, 16, 20, 26] with the references therein.

In our Matlab package we have added, for the purpose of comparison, the routines that provide a compressed Tchakaloff-like rule along the lines sketched above, starting from a tetrahedralization of the polyhedral element. We notice that even this approach does not seem to be yet readily available, at least in Matlab, within existing open-source packages, and thus could be of independent practical interest, especially because it resorts to the best tetrahedral rules available in the literature.

3. Numerical tests

In this section we test Algorithm TetraFreeQ on several polyhedral domains. All the routines have been implemented in Matlab and are freely available at [19]. The numerical experiments were made using Matlab R2022a, on an Apple MacBook Pro with M1 Chip and 16 GB of RAM.

We consider three polyhedra Ω_j , $j = 1, 2, 3$ with triangular facets, where Ω_1 is nonconvex with 30 facets, Ω_2 is convex (an approximation of a sphere with 760 facets) and Ω_3 is multiply-connected with 20 facets and a hole (see Figure 1). The domains, as well as the facets and the tetrahedralization (used for comparison), are obtained by Matlab built-in command `alphashape`, on suitable point clouds.

In particular we compute on the three polyhedra:

- a reference rule Q_T corresponding to a tetrahedralization (cf. Remark 2.2)
- the corresponding compressed tetrahedral rule Q_{TC}
- the tetrahedra-free rule Q_{TF} produced by Algorithm TetraFreeQ.

In our numerical tests, in (ii1)–(ii4) of Algorithm TetraFreeQ we have chosen the parameters $\varepsilon = 10^{-14}$, $\theta = 4$ and an initial value $K = \lceil 8N \text{vol}(B) / \text{vol}(\Omega) \rceil$ where B is the smallest bounding box for Ω .

In order to test numerically polynomial exactness of Q_{TF} , in Figure 3 we display for each domain the relative errors

$$E(g_k) = \frac{|Q_T(g_k) - Q_{TF}(g_k)|}{|Q_T(g_k)|} \quad (14)$$

corresponding to 100 trials of the polynomial

$$g_k(x, y, z) = (a_k + b_k x + c_k y)^n \quad (15)$$

where n is the algebraic degree of precision of the rule and $(a_k, b_k, c_k) \in [0, 1]$ are uniform random coefficients. In the same figures, we have also represented with a circle the logarithmic averages $\sum_{k=1}^{100} \log(E(g_k)) / 100$.

As additional investigation, we also compute the relative errors of these rules in integrating on Ω_j , $j = 1, 2, 3$, the three test functions with different degree of regularity

$$f_1(x, y, z) = \exp(-\|P - P_0\|_2^2), \quad f_2(x, y, z) = \|P - P_0\|_2^5, \quad f_3(x, y, z) = \|P - P_0\|_2, \quad (16)$$

where $P = (x, y, z)$ and $P_0(x_0, y_0, z_0)$ was suitably chosen in the domains, more precisely $P_0 = (1.5, 1.5, 1.5)$ for Ω_1 and Ω_3 , $P_0 = (1, 1, 1)$ for Ω_2 ; cf. Figures 4-6. Moreover, in Table 2 we report the cardinalities of the pointsets from which the tetrahedra-free rules are extracted, varying the degree from 1 to 10.

In view of these numerical experiments, we can see that the rules Q_{TC} and Q_{TF} give errors of comparable size, whereas the errors of Q_T are smaller. This is not surprising, since the basic tetrahedra-based rule uses a much larger number of nodes than N , which turns out to be the number of nodes of both Q_{TC} and Q_{TF} , with a ratio increasing with the number of facets (being for example more than 100 with the 760-facets sphere-like polyhedron, cf. Figure 2). We stress once again, however, that Q_{TF} avoids completely any tetrahedralization of the polyhedron, a desirable feature in many applications (such as polyhedral FEM/VEM methods).

Concerning the cputime required by the moment computation in TetraFreeQ, it grows with the algebraic degree of precision and of course with the number of facets, ranging in our tests between 10^{-4} and 10^{-3} seconds. On the other hand, there is numerical evidence that the bottleneck of TetraFreeQ is due to the in-polyhedron routine and the NNLS algorithm, one dominating on the other depending on the algebraic degree of exactness and on the number of facets, with an overall cputime of the quadrature rule construction varying from 10^{-2} seconds for $n = 1, 2, 3$ up to 10^0 seconds for $n = 10$.

4. Acknowledgements

Work partially supported by the DOR funds and the biennial project BIRD 192932 of the University of Padova, and by the INdAM-GNCS 2022 Project “Methods and software for multivariate integral models”. This research has been accomplished within the RITA “Research ITalian network on Approximation”, and within the UMI Group TAA “Approximation Theory and Applications” (A. Sommariva).

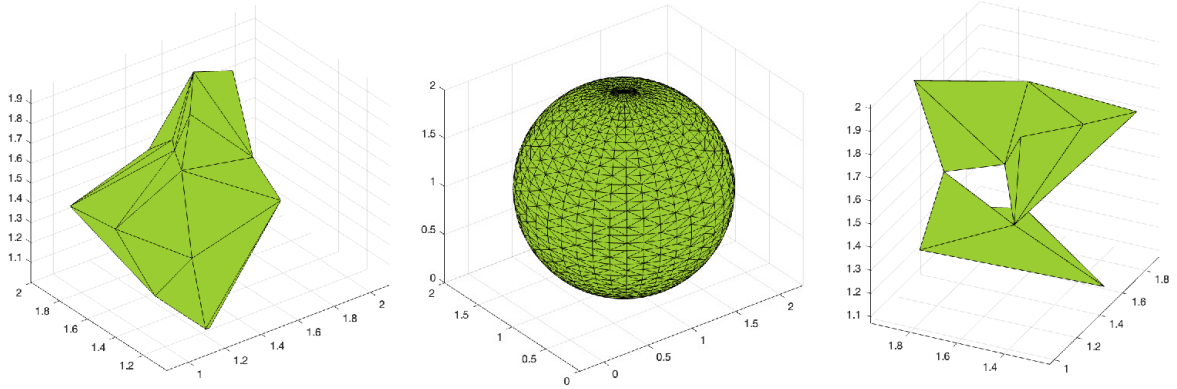


Figure 1: Examples of polyhedral domains. Left: Ω_1 (nonconvex, 20 facets); Center: Ω_2 (convex, 760 facets); Right: Ω_3 (multiply connected, 20 facets).

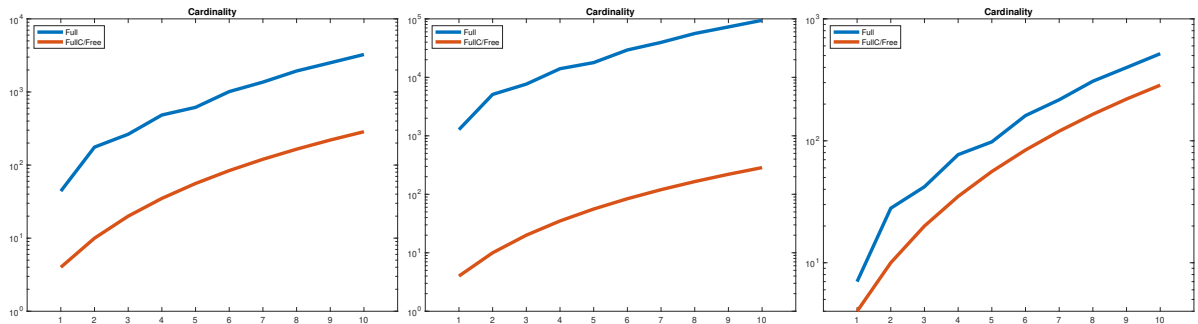


Figure 2: Cardinality versus the exactness degree on the three polyhedra of Figure 1: uncompressed tetrahedra-based rule (blue), compressed tetrahedra-based and tetrahedra-free rule (brown).

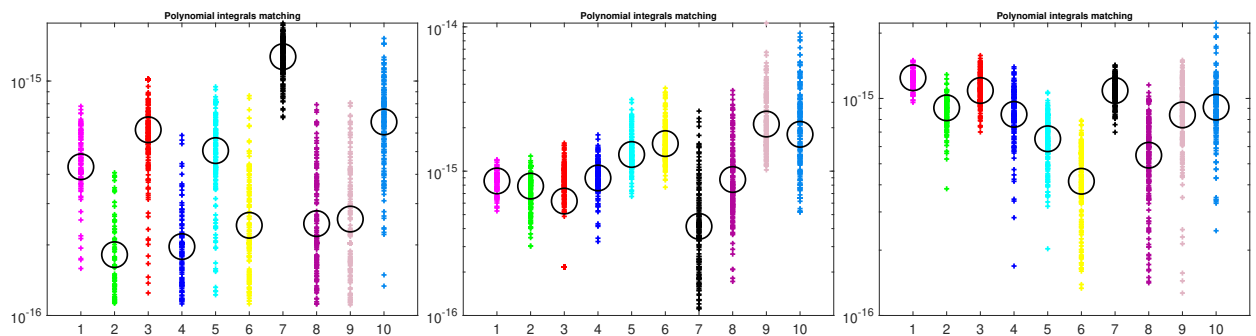


Figure 3: Relative errors $E(g_k)$ of the tetrahedra-free rule over 100 polynomial integrands of the form $g_k = (a_k x + b_k y + c_k z + d_k)^n$ on the three polyhedra of Figure 1, where a_k, b_k, c_k, d_k are uniform random coefficients in $[0, 1]$ and $n = 1, 2, \dots, 10$; the circles correspond to the average logarithmic error $\sum_{k=1}^{100} \log(E(g_k))/100$.

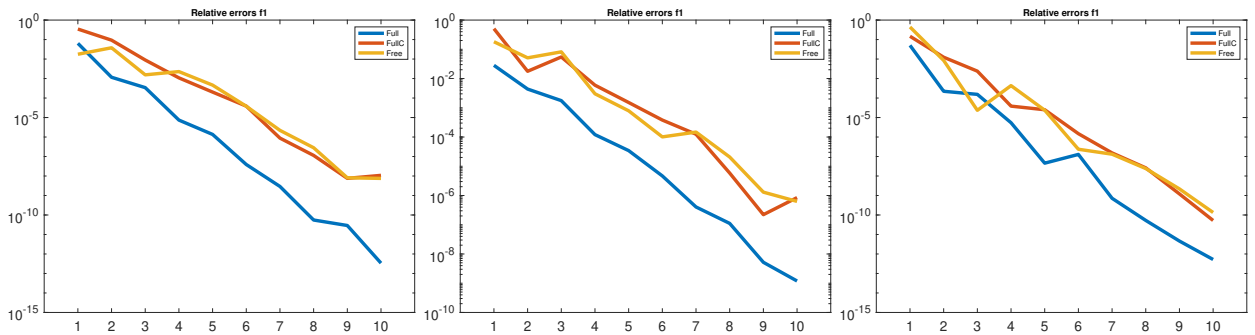


Figure 4: Relative errors versus the exactness degree in the integration of the Gaussian f_1 in (16) on the three polyhedra of Figure 1: uncompressed (blue) and compressed (brown) tetrahedra-based rules, tetrahedra-free rule(yellow).

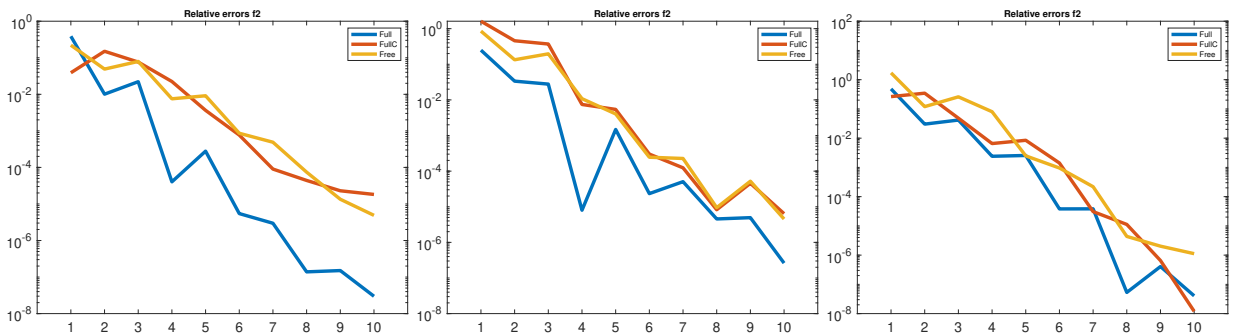


Figure 5: Relative errors versus the exactness degree in the integration of the power function f_2 in (16) on the three polyhedra of Figure 1: uncompressed (blue) and compressed (brown) tetrahedra-based rules, tetrahedra-free rule(yellow).

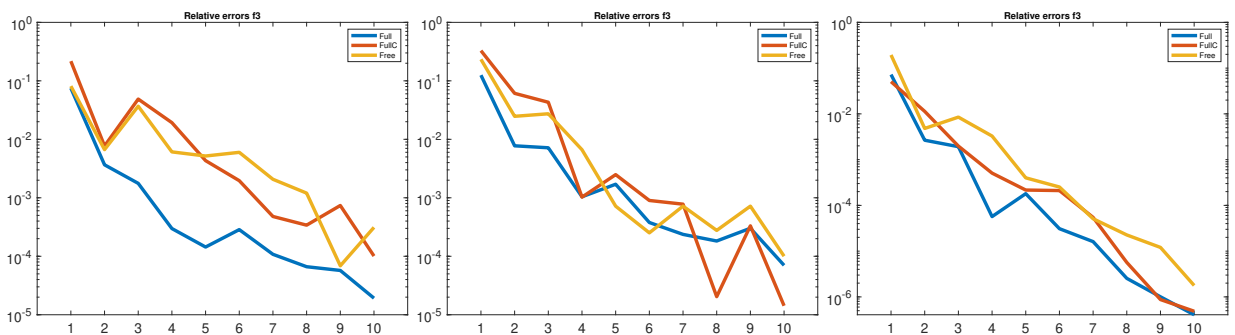


Figure 6: Relative errors versus the exactness degree in the integration of the distance function f_3 in (16) on the three polyhedra of Figure 1: uncompressed (blue) and compressed (brown) tetrahedra-based rules, tetrahedra-free rule(yellow).

deg	1	2	3	4	5	6	7	8	9	10
N	4	10	20	35	56	84	120	165	220	286
Ω_1	31	77	160	286	447	676	969	1327	1765	2286
Ω_2	28	78	154	273	450	669	956	1313	1757	2287
Ω_3	29	77	156	276	446	662	950	1312	1753	2285

Table 2: Cardinality of the Halton pointsets from which tetrahedra-free rules with N nodes are obtained.

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*, 2nd ed., vol. II, Blaisdell, 1969.
- [3] B. Bauman, A. Sommariva, M. Vianello, Compressed cubature over polygons with applications to optical design, *J. Comput. Appl. Math.*, 370, 2020.
- [4] L. Beirão da Veiga, F. Dassi, A. Russo, High-order Virtual Element Method on polyhedral meshes, *Comput. Math. Appl.* 74 (2017) 1110–1122.
- [5] M. Dell’Orto, M. Dessolet, F. Marcuzzi, The Lawson-Hanson Algorithm with Deviation Maximization: Finite Convergence and Sparse Recovery, *Numer. Linear Algebra Appl.*, accepted upon revision. arXiv:2108.05345v3, 2022.
- [6] M. Dessolet, F. Marcuzzi, Deviation maximization for rank-revealing QR factorizations, *Numer. Algorithms* 91 (2022) 1047–1079.
- [7] M. Dessolet, F. Marcuzzi, M. Vianello, Accelerating the Lawson-Hanson NNLS solver for large-scale Tchakaloff regression designs, *Dolomites Res. Notes Approx. DRNA* (2020) 20–29.
- [8] P.J. Davis, A construction of nonnegative approximate quadratures, *Math. Comp.* 21 (1967) 578–582.
- [9] S. Hayakawa, Monte Carlo cubature construction, *Jpn. J. Ind. Appl. Math.*, 38 (2021) 561–577.
- [10] J.S. Hesthaven, G. Rozza, B. Stamm, *Certified reduced basis methods for parametrized partial differential equations*, Springer, New York, 2016.
- [11] S. Holcombe, INPOLYHEDRON - are points inside a triangulated volume?, 2015, <https://www.mathworks.com/matlabcentral/fileexchange/37856-inpolyhedron-are-points-inside-a-triangulated-volume>.
- [12] J. Jaskowicz, N. Sukumar, High-order cubature rules for tetrahedra, *Int. J. Numer. Methods Eng.* 121 (2020) 2418–2436.
- [13] C.L. Lawson, R.J. Hanson, *Solving least squares problems*, Classics in Applied Mathematics 15, SIAM, Philadelphia, 1995.
- [14] The Mathworks, *alphaShape*, Polygons and polyhedra from points in 2-D and 3-D, <https://www.mathworks.com/help/matlab/ref/alphashape.html>.
- [15] S.E. Mousavi, N. Sukumar, Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons, *Comput. Mech.* 47 (2011) 535–554.
- [16] F. Piazzon, A. Sommariva, M. Vianello, Caratheodory-Tchakaloff Subsampling, *Dolomites Res. Notes Approx. DRNA* 10 (2017) 5–14.
- [17] E. Schönhardt, Über die Zerlegung von Dreieckspolyedern in Tetraeder. *Mathematische Annalen*, 98:309–312, 1928.
- [18] L. Shunn, F. Ham, Symmetric quadrature rules for tetrahedra based on a cubic close-packed lattice arrangement, *J. Comput. Appl. Math.* 236 (2012) 4348–4364.
- [19] A. Sommariva, *Quadrature and Cubature Codes*, available at <https://www.math.unipd.it/~alvise/software.html>.
- [20] A. Sommariva, M. Vianello, Compression of multivariate discrete measures and applications, *Numer. Funct. Anal. Optim.* 36 (2015) 1198–1223.
- [21] A. Sommariva, M. Vianello, Computing Tchakaloff-like cubature rules on spline curvilinear polygons, *Dolomites Res. Notes Approx. DRNA* 14 (2021) 1–11.
- [22] A.H. Stroud, *Approximate Calculation of Multiple Integrals*, Prentice-Hall, Englewood Cliffs, 1971.
- [23] Y. Sudhakar, J.P.M. De Almeida, W.A. Wall, An accurate, robust, and easy-to-implement method for integration over arbitrary polyhedra: application to embedded interface methods, *J. Comput. Phys.* 273 (2014), 393–415.
- [24] Y. Sudhakar, A. Sommariva, M. Vianello, W.A. Wall, On the use of compressed polyhedral quadrature formulas in embedded interface methods, *SIAM J. Sci. Comput.* 39 (2017) B571–B587.
- [25] V. Tchakaloff, Formules de cubatures mécaniques à coefficients non négatifs, (French), *Bull. Sci. Math.* 81 (1957) 123–134.
- [26] M. Tchernychova, *Caratheodory cubature measures*, Ph.D. dissertation in Mathematics (supervisor: T. Lyons), University of Oxford, 2015.
- [27] E.W. Weisstein, *Polygon Area*, from Wolfram MathWorld, <https://mathworld.wolfram.com/PolygonArea.html>.
- [28] D.R. Wilhelmsen, A Nearest Point Algorithm for Convex Polyhedral Cones and Applications to Positive Linear approximation, *Math. Comp.* 30 (1976) 48–57.