Tchakaloff-like polyhedral quadrature with and without tetrahedralization

Alvise Sommariva and Marco Vianello

Functional Analysis, Approximation Theory and Numerical Analysis
Matera, Italy, July 5-8, 2022

Thanks: RITA, UMI group TAA, GNCS project *Metodi e software per la modellistica integrale multivariata*

We present an algorithm that computes an algebraic cubature rule

$$\int_{\Omega} f(x,y,z)dxdydz \approx \sum_{j=1}^{\eta} w_j f(Q_j)$$

over general polyhedra $\Omega \subset \mathbb{R}^3$.

- The motivation is the lack for available routines in Matlab.
- The intention is to provide algorithms with and without tetrahedralization.
- The degrees δ are *mild* (say less than 10).

This approach is well-known in literature.

- Determine a triangulation $\mathcal{T} = \{T_k\}_{k=1,\dots,M}$ of the polyhedron Ω, i.e. $\Omega = \cup_{k=1}^{M} T_k$ and the intersection of the interior of two distinct tetrahedrons T_k is empty.
- Compute the integral $Q_{\delta}^{(k)}(f) = \sum_{j=1}^{N_k} w_j^{(k)} f(P_j^{(k)})$ by a rule with algebraic degree of exactness δ on each T_k, $k = 1, \dots, M$.
- In view of the additivity of the integration operator we get a rule of degree δ on Ω, i.e.

$$I_{\Omega}(f) \approx \sum_{k=1}^{M} Q_{\delta}^{(k)}(f) = \sum_{k=1}^{M} \sum_{j=1}^{N_k} w_j^{(k)} f(P_j^{(k)}).$$

Some considerations about the **triangulation**.

- If the polyhedron Ω is not convex/star shaped (knowing a center!), the determination of the triangulation may not be straightforward.
- If Ω is obtained by `alphashape` from a point cloud of vertices, the command `alphaTriangulation` returns a triangulation of Ω.
- Note that by varying the `alphashape` parameter, the obtained domain can be very different.

Some considerations about the **rules on the tetrahedra** with internal nodes and positive weights.

For degrees of precision $\delta = 0, 1, \ldots, 20$, there are in literature several pointsets that are exact for all the polynomials of total degree δ on the reference tetrahedron T^* with vertices $[1, 0, 0]$, $[0, 1, 0]$, $[0, 0, 0]$, $[0, 0, 1]$ and have almost-minimal cardinality.

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: Cardinality of almost-minimal rules on reference tetrahedron.

- All these rules have internal nodes and positive weights.
- For $\delta > 20$, one can use a the well-established Stroud rule, that in general has a not minimal cardinality but it is easy to be implemented.

- Once a rule is available on the reference tetrahedron T^*, it can be easily obtained on each T_k by barycentric coordinates and the computation of T_k volume.
- If the cardinality L of the rule on the wanted polyhedron Ω is higher than

$$\tilde{L}_\delta = (\delta+1)(\delta+2)(\delta+3)/6$$

then one can extract a Tchakaloff rule with at most $\tilde{L}_n$ internal nodes and positive weights by means of Lawson-Hanson algorithm. This process is fast for mild δ.

Alternatively one can apply a QR approach, that is faster but does not guarantee the positiveness of the weights.

The procedure works essentially as follows:

- we compute the moments $\{\gamma_k\}_{k=1,\dots,N}$ of a certain polynomial basis $\{\phi_k\}_{k=1,\dots,N}$ of tensorial type by means of cubature rules with ADE $\delta+1$ on the polyhedron facets $\{\mathcal{F}_i\}_{i=1,\dots,M}$, in virtue of the divergence theorem;
- using an `inpolygon` routine we consider a sufficient number of points $\{\tilde{P}_l\}_{l=1,\dots,L}$ inside Ω so that the overdetermined linear system $V'w=\gamma$, with $V_{l,k}=(\phi_k(\tilde{P}_l))$, has a nonnegative solution w with at most $N \leq L$ positive components.
- extract a rule with positive weights and internal nodes via fast Lawson-Hanson algorithm.

In spite of the simplicity of this approach there are many aspects that deserve explanations, on the implementation side as well as on the theoretical one.

Algorithms without tetrahedralization: moment computation

Some considerations about the **moment computation**.

For all the triples $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ with $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{N}$ and $\alpha_1 + \alpha_2 + \alpha_3 \leq \delta$, one must compute the moments of the tensorial Chebyshev basis on the bounding box of Ω, i.e.

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

where, being T_m the Chebyshev polynomial of first kind, of degree m,

$$\tilde{T}_m^{(a,b)}(t) := T_m\left(\left(x - \frac{a+b}{2}\right)\frac{2}{b-a}\right). \tag{1}$$

One can show that in view of divergence theorem it is equivalent to compute

$$\gamma_\alpha = \sum_{k=1}^{M} \int_{\mathcal{F}_k} n_1^{(k)}\; U_{\alpha_1}^{(a_1,b_1)}(x)\tilde{T}_{\alpha_2}^{(a_2,b_2)}(y)\tilde{T}_{\alpha_3}^{(a_3,b_3)}(z)\; dS$$

where $\mathcal{F}_k$ are the polyhedra facets with outer normals $n_1^{(k)}$ and $U_{\alpha_1}^{(a,b)} \in \mathbb{P}_{\alpha_1+1}$

$$U_0^{(a,b)}(x) = x - \frac{a+b}{2},\; U_1^{(a,b)}(x) = \frac{1}{b-a}\left(x - \frac{a+b}{2}\right)^2, U_m^{(a,b)} = \frac{2}{b-a}\left(\frac{\tilde{T}_{m+1}^{(a,b)}(x)}{2(m+1)} - \frac{\tilde{T}_{m-1}^{(a,b)}(x)}{2(m-1)}\right),$$

Since

$$\gamma_\alpha = \sum_{k=1}^{M} \int_{\mathcal{F}_k} n_1^{(k)} \; U_{\alpha_1}^{(a_1,b_1)}(x)\tilde{T}_{\alpha_2}^{(a_2,b_2)}(y)\tilde{T}_{\alpha_3}^{(a_3,b_3)}(z) \; dS$$

we compute the k-th term of the sum by a cubature rule on the polygonal facet $\mathcal{F}_k$.

Note that each integrand is a polynomial of total degree at most $\delta + 1$.

By an affine map, with some care, this can be conveniently done by cubature of degree $\delta + 1$ over a suitable planar polygon $\mathcal{F}_k^{(2)} \subset \mathbb{R}^2$, thing that can be done even without triangulations of $\mathcal{F}_k^{(2)}$.

Finally we can compute a **cubature formula** with positive weights and internal nodes as follows.

1. generate a set of random-points $\mathcal{P}^{(1)} = \{P_i\}_{i=1}^{k_1}$ in the smallest parallelepiped $[a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ containing Ω (k_1 well-choosen!);
2. determine those points $\mathcal{P}^{(2)} = \{P_j^{(2)}\}_{j=1}^{k_2} \subseteq \mathcal{P}^{(1)}$ belonging to Ω (e.g. by open-source routine `inpolyhedron`);
3. by a procedure of Lawson-Hanson type, for instance using Matlab built-in `lsqnonneg` or the alternative open source `LHDM`,
   - extract a set of nodes $\mathcal{Q}^{(1)} = \{Q_j\}_{j=1}^{k_3} \subseteq \mathcal{P}^{(2)}$,
   - compute the relative (positive) weights $\{w_j\}_{j=1}^{k_3} \subseteq \mathbb{R}^+$,

   so that the moment error $\|V^T w - \gamma_k\|_2$ is less than `tol`, where $V_{i,j} = \psi_j(P_i^{(2)})$ and $\gamma_k = \int_\Omega \psi_k(x, y, z)\ dx\, dy\, dz$ (ψ_k is the $\mathbb{P}_\delta$ basis on the smallest parallelepiped) and `tol` is a tolerance fixed by the user, e.g. `tol`$=10^{-14}$;
4. in case of failure, generate new random-points, and restart from item 1, also using the already defined internal points $\mathcal{P}^{(2)}$.

Fundamental: a result by Wilhelmsen says that in theory this procedure will have success for sufficiently dense data.

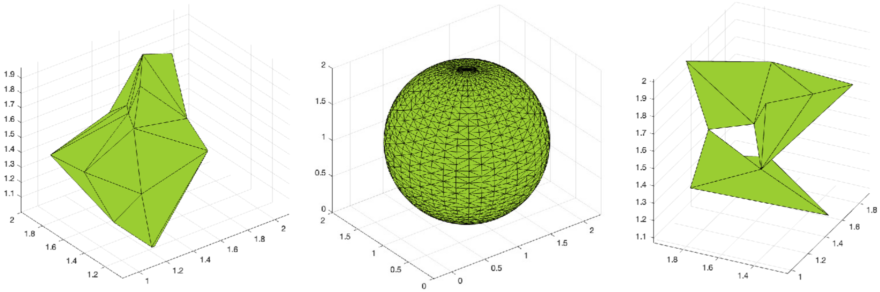

Figure: Examples of polyhedral domains.

Left: non convex, Center: convex, Right: non convex with hole.

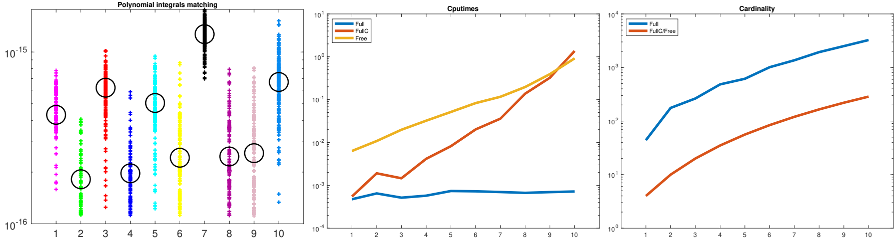

Figure: Domain 1 (30 facets): Moment matching of the free/not free methods over 100 integrands of the form $(c_1 + k_1 \cdot x + \cdot y + k_3 \cdot z)^{\delta}$ where $c_1, k_1, k_2, k_3 \in [0, 1]$ are random, average cputime and cardinality. Triangulation cputime: $5e - 3$ seconds.

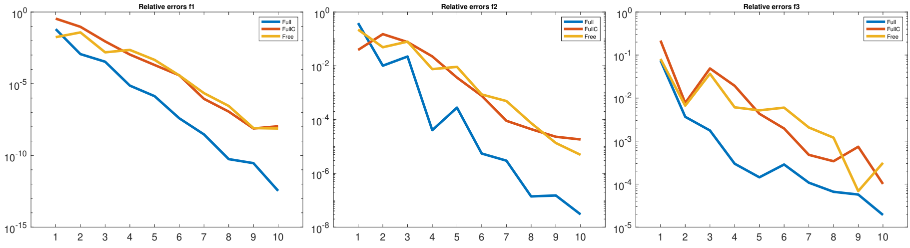

Figure: Domain 1 (30 facets): Relative errors integrating $f_1(x,y,z) = \exp(-x^2 - y^2 - z^2)$,
$f_2(x,y,z) = ((x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2)^{5/2}$,
$f_3(x,y,z) = ((x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2)^{1/2}$, con $(x_0, y_0, z_0) = (1.5, 1.5, 1.5)$.

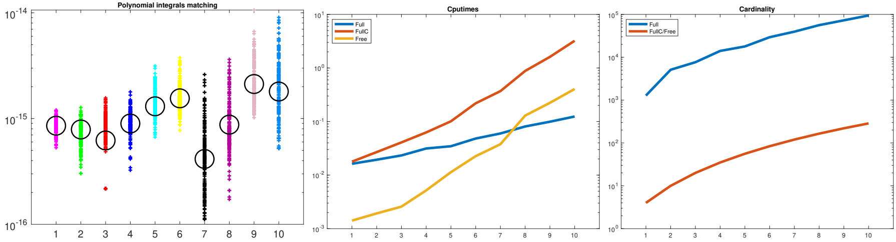

Figure: Domain 2 (760 facets, sphere like): Moment matching of the free/not free methods over 100 integrands of the form $(c_1 + k_1 \cdot x + \cdot y + k_3 \cdot z)^\delta$ where $c_1, k_1, k_2, k_3 \in [0, 1]$ are random, average cputime and cardinality. Triangulation cputime: $8e - 2$ seconds. The indomain is fast since the domain is convex.

Numerical experiments: domain 2, integration of some functions

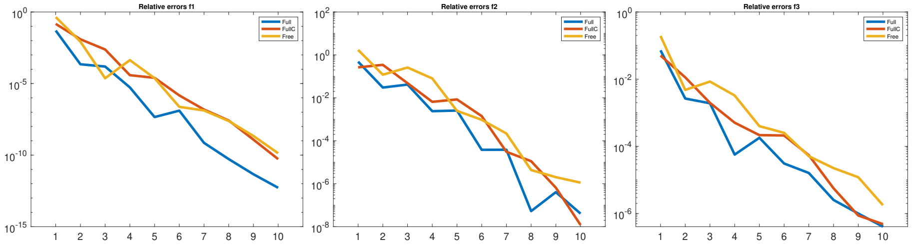

Figure: Domain 2 (760 facets, sphere like): Relative errors integrating $f_1(x,y,z) = \exp(-x^2 - y^2 - z^2)$, $f_2(x,y,z) = ((x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2)^{5/2}$, $f_3(x,y,z) = ((x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2)^{1/2}$, con $(x_0, y_0, z_0) = (1,1,1)$.

Numerical experiments: domain 3

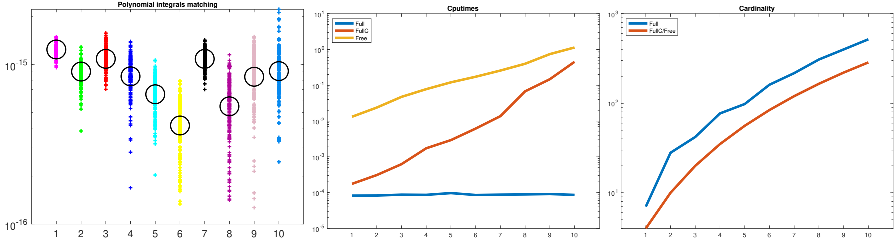

Figure: Domain 3 (20 facets, hole): Moment matching of the free/not free methods over 100 integrands of the form $(c_1 + k_1 \cdot x + \cdot y + k_3 \cdot z)^{\delta}$ where $c_1, k_1, k_2, k_3 \in [0, 1]$ are random, average cputime and cardinality. Triangulation cputime: $5e - 3$ seconds.

Numerical experiments: domain 2, integration of some functions

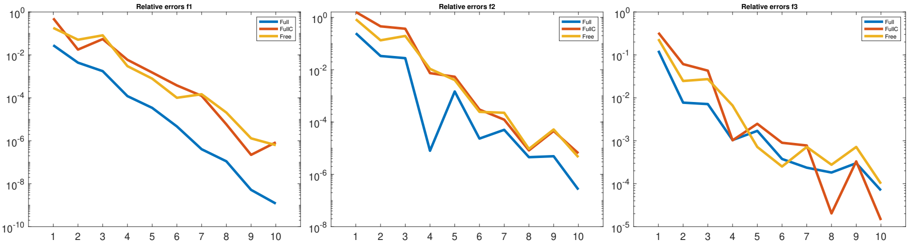

Figure: Domain 3 (20 facets, hole): Relative errors integrating $f_1(x,y,z) = \exp(-x^2 - y^2 - z^2)$, $f_2(x,y,z) = ((x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2)^{5/2}$, $f_3(x,y,z) = ((x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2)^{1/2}$, con $(x_0, y_0, z_0) = (1.5, 1.5, 1.5)$.

Our intention is to propose a fast and reliable code. In this sense we intend to

- find faster indomain routines for polyhedra;
- find faster Lawson-Hanson method (collaborators work in progress);
- find best parameters (e.g., fewer points from which extract the final nodes);
- many more stress tests for the routines;
- application to PDE problems.

In terms of cputime there is no problem with the moment computation (it is fast and accurate).

Important: All the Matlab routines will be available at the authors' homepages.

Bibliography

1. P.F. Antonietti, P. Houston, G.Pennesi, *Fast Numerical Integration on Polytopic Meshes with Applications to Discontinuous Galerkin Finite Element Methods*. J. of Scientific Comp. 77, 1339–1370 (2018).

   Quadratures without the need to partition the domain into triangles or tetrahedrons, no software.

2. M. Dessole, F. Marcuzzi, M. Vianello, *Accelerating the Lawson-Hanson NNLS solver for large-scale Tchakaloff regression designs*, DRNA 13, 20–29 (2020).

   Fast Lawson-Hanson algorithm and Matlab codes.

3. S. Holcombe, *Matlab open source routine:* `inpolyhedron`

   Open source Matlab in-domain routine for polyhedra.

4. S.E. Mousavi, N. Sukumar, *Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons*. Comput. Mech. 47, 5 (2011), 535–554.

   Cubature without triangulations, no software. Moments by Lasserre's method.

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

   (Details on cubature compression)

6. D. R. Wilhelmsen, A Nearest Point Algorithm for Convex Polyhedral Cones and Applications to Positive Linear approximation, *Math. Comp.*, 30 (1976), 48-57.