Numerical cubature over polygons and polyhedra

Alvise Sommariva

Maynooth (Ireland), Wednesday, April 29, 2026

- In this talk we report some recent methods to perform numerical cubature over polygons and polyhedra.
- We start our description introducing algorithms that compute low-cardinality quadrature rules with a given degree of polynomial exactness, using triangulations of the domains.
- Later we show how we can bypass this assumptions by means of meshfree methods, that have theoretical bases on Tchakaloff and Davis-Wilhelmsen theorem.
- Time permitting, we conclude the talk by showing a new technique named Orthocub that is implemented via orthogonal polynomial moments and auxiliary near-minimal algebraic cubature in a bounding box of the domain, with no conditioning issue since no matrix inversion or factorization is needed.

We intend to numerically approximate

$$\int_{\Omega} f(x) d\Omega \approx \sum_{i=1}^{N_\delta} w_i f(P_i).$$

where

- Ω is a domain, e.g. an interval, a polygon or a polyhedron,
- $f \in C(\Omega)$.

by a formula that has algebraic degree of exactness δ, that is

$$\int_{\Omega} f(x) d\Omega = \sum_{i=1}^{N_\delta} w_i f(P_i), \quad f \in \mathbb{P}_\delta$$

$\mathbb{P}_\delta$ being the set of algebraic polynomials of *total* degree at most δ.

The points P_i, $i = 1, \ldots, N_\delta$ are the nodes, while w_i are the weights.

Remark

Later, ADE is the algebraic degree of exactness δ of the formula.

In the formula

$$\int_{\Omega} f(x) d\Omega \approx \sum_{i=1}^{N_\delta} w_i f(P_i)$$

depending on the applications one may require that

- the nodes P_i, $i = 1, \ldots, N_\delta$ are all internal to the domain Ω since the function f can be evaluated only in Ω;
- some of the nodes P_i, $i = 1, \ldots, N_\delta$ can be external to the domain Ω since the function f can be evaluated also outside Ω (e.g. f is a polynomial);
- all the weights w_i, $i = 1, \ldots, N_\delta$ are positive so that the r.h.s. can be seen as an integral w.r.t. a discrete positive measure;
- some of the weights can be negative, since independently of the theoretical view this assumption does not hurt numerically.

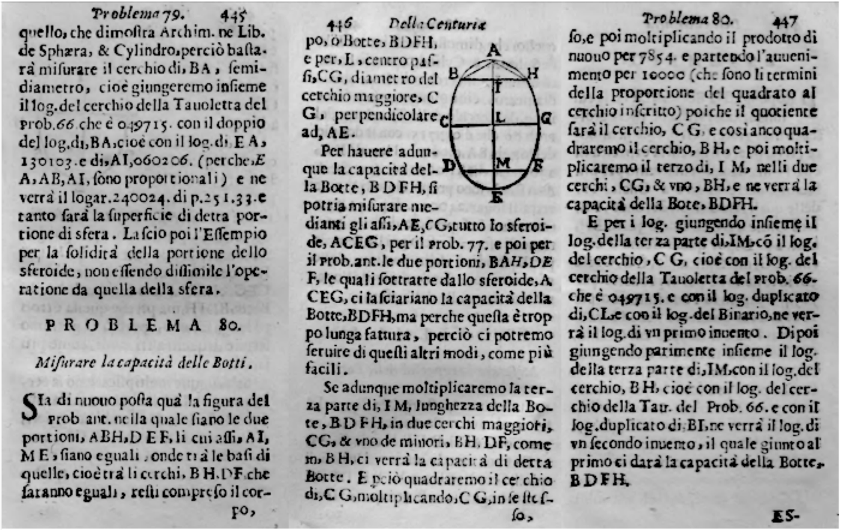
In general for numerical issues, if the rule has the algebraic degree of exactness ADE at least $\delta = 0$ (the formula integrates exactly the constants) the index of stability named conditioning of the cubature formula

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

is required to be close to 1.

- The optimal value of $\mathrm{cond}(\{w_i\})$ is obtained by rules with positive weights and is equal to 1.
- In case the rule has negative weights, then $\mathrm{cond}(\{w_i\})$ is strictly larger than 1. In general one requires that $\mathrm{cond}(\{w_i\})$ is not too large (say smaller than 2 or 3).

- Numerical integration over intervals has a long history, from the Newton-Cotes rules on bounded intervals, to their composite versions.
- Particularly interesting are those Newton-Cotes rules with 2 points (trapezoidal rule) and 3 points (Cavalieri-Simpson rule).
- In general the Newton-Cotes rules have ADE at least equal to $N_\delta - 1$, where N_δ is the number of nodes and for stability issues it is not suggested to choose $N_\delta \geq 7$, while alternatively composite versions of trapezoidal or Cavalieri-Simpson rule are very used in practice (see e.g. the paper by Trefethen The Exponentially Convergent Trapezoidal Rule).



Problema 79. 445

quello, che dimoſtra Archim. ne Lib. de Sphæra, & Cylindro, perciò baſterà miſurare il cerchio di, BA, ſemidiametro, cioè giungeremo inſieme il log. del cerchio della Tauoletta del prob. *66* che è 049715. con il doppio del log. di, BA, cioè con il log. di, E A, 130103. e di, AI, 060206. (perche, *E* A, AB, AI, ſono proportionali) e ne verrà il logar. 240024. di p. 251.33. e tanto ſarà la ſuperficie di detta portione di sfera. Laſcio poi l'Eſſempio per la ſolidità della portione dello sferoide, non eſſendo diſſimile l'operatione da quella della sfera.

P R O B L E M A 80.

Miſurare la capacità delle Botti.

SIa di nuouo poſta quà la figura del prob. ant. nella quale ſiano le due portioni, ABH, D E F, li cui aſſi, AI, M E, ſiano eguali, onde trà le baſi di quelle, cioè trà li cerchi, B H, DF, che ſaranno eguali, reſti compreſo il cor-

po,

446 *Della Centuria*

po, ò Botte, BDFH, e per, L, centro paſſi, CG, diametro del cerchio maggiore, C G, perpendicolare ad, AE.

Per hauere adunque la capacità della Botte, B D F H, ſi potria miſurare mediant gli aſſi, AE, CG, tutto lo sferoide, ACEG, per il prob. 77. e poi per il prob. ant. le due portioni, BAH, *DE* F, le quali ſottratte dallo sferoide, A CEG, ci laſciariano la capacità della Botte, BDFH, ma perche queſta è troppo lunga fattura, perciò ci potremo ſeruire di queſti altri modi, come più facili.

Se adunque moltiplicaremo la terza parte di, I M, lunghezza della Botte, B D F H, in due cerchi maggiori, CG, & vno de minori, BH, DF, come in, B H, ci verrà la capacità di detta Botte. E però quadraremo il cerchio di, C G, moltiplicando, C G, in ſe ſteſ-

ſo,

Problema 80. 447

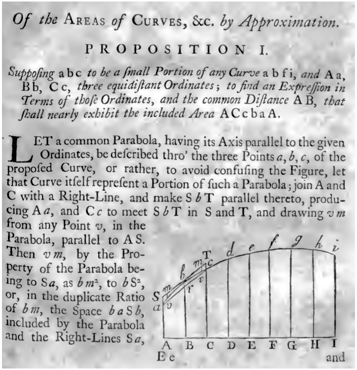
ſo, e poi moltiplicando il prodotto di nuouo per 7854. e partendo l'auuenimento per 10000 (che ſono li termini della proportione del quadrato al cerchio inſcritto) poiche il quotiente ſarà il cerchio, C G, e coſi anco quadraremo il cerchio, B H, e poi moltiplicaremo il terzo di, I M, nelli due cerchi, CG, & vno, BH, e ne verrà la capacità della Bote, BDFH.

E per i log. giungendo inſieme il log. della terza parte di, IM, cõ il log. del cerchio, C G, cioè con il log. del cerchio della Tauoletta del prob. *66.* che è 049715. e con il log. duplicato di, CL, e con il log. del Binario, ne verrà il log. di vn primo inuento. Dipoi giungendo parimente inſieme il log. della terza parte di, IM, con il log. del cerchio, B H, cioè con il log. del cerchio della Tau. del Prob. 66. e con il log. duplicato di, BI, ne verrà il log. di vn ſecondo inuento, il quale giunto al primo ci darà la capacità della Botte, BDFH.

ES-

Figure: Problem 80, at p.445 of *Centuria di varii problemi* by Cavalieri.

The same rule is available at Mathematical dissertations On A Variety of Physical and Analytical subjects by Thomas Simpson, adding a path towards composite rules.

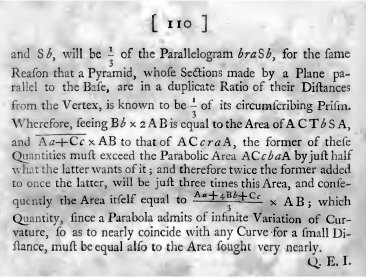


Of the AREAS *of* CURVES, &c. *by Approximation.*

PROPOSITION I.

Suppoſing abc *to be a ſmall Portion of any Curve* abfi, *and* Aa, Bb, Cc, *three equidiſtant Ordinates; to find an Expreſſion in Terms of thoſe Ordinates, and the common Diſtance* AB, *that ſhall nearly exhibit the included Area* ACcbaA.

LET a common Parabola, having its Axis parallel to the given Ordinates, be deſcribed thro' the three Points *a*, *b*, *c*, of the propoſed Curve, or rather, to avoid confuſing the Figure, let that Curve itſelf repreſent a Portion of ſuch a Parabola; join A and C with a Right-Line, and make S*b*T parallel thereto, producing A*a*, and C*c* to meet S*b*T in S and T, and drawing *vm* from any Point *v*, in the Parabola, parallel to AS. Then *vm*, by the Property of the Parabola being to S*a*, as bm^2, to bS^2, or, in the duplicate Ratio of *bm*, the Space *ba*S*b*, included by the Parabola and the Right-Lines S*a*, and



[110]

and S*b*, will be $\frac{1}{3}$ of the Parallelogram *bra*S*b*, for the ſame Reaſon that a Pyramid, whoſe Sections made by a Plane parallel to the Baſe, are in a duplicate Ratio of their Diſtances from the Vertex, is known to be $\frac{1}{3}$ of its circumſcribing Priſm. Wherefore, ſeeing B*b* × 2AB is equal to the Area of ACT*b*SA, and $\overline{Aa+Cc} \times AB$ to that of AC*cra*A, the former of theſe Quantities muſt exceed the Parabolic Area AC*cba*A by juſt half what the latter wants of it; and therefore twice the former added to once the latter, will be juſt three times this Area, and conſequently the Area itſelf equal to $\frac{Aa+4Bb+Cc}{3} \times AB$; which Quantity, ſince a Parabola admits of infinite Variation of Curvature, ſo as to nearly coincide with any Curve for a ſmall Diſtance, muſt be equal alſo to the Area ſought very nearly. Q. E. I.

Figure: Dal testo di Simpson *Mathematical dissertations On A Variety of Physical and Analytical subjects.*

[1] Phrasi de Cavalieri es « Per havére la capacità della botte moltiplicaremo la terza parte della lunghezza della botte in due cerchi maggiori ed uno dei minori ».

Versione « Pro habe capacitate de vase vinario, nos multiplica tercio parte de longitudine de vase per duo circulo maiore et uno minore ».

Si nos sume axi de vase pro axi de x, et pro origine puncto medio, si longitudine de vase es h, et si $f(x)$ es area de sectione de vase, in puncto de abscissa x, et normale ad axi, tunc

volumine $= \int_{-h/2}^{h/2} f(x)\,dx$; circulo maiore $= f(0)$, circulo minore $= f(h/2 = f(-h/2)$; et regula de Cavalieri dic que integrale vale :

$$\frac{h}{3}[2f(0) + f(h/2)] = \frac{h}{6}[4f(0) + f(-h/2) + f(h/2)] \ .$$

Figure: A note by Peano on the work of Cavalieri.

- A great discovery has been that of Gaussian rules that allow the approximation of

$$\int_a^b f(x)w(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

for a weight function w, even on unbounded intervals (a, b).

- The nodes $\{x_i\}$ are all internal to the interval (a, b), the weights w_i are positive and the ADE δ is equal to $2n-1$, that turns out to be maximum possible with n nodes.

- In general numerical routines must compute the nodes and the weights. This purpose in the sixties made available these quantities for low/mild n but now one can compute them for several thousands to even millions, very rapidly.

The accompanying table computed by the Mathematical Tables Project gives the roots x_i for each $P_n(x)$ up to $n=16$, and the corresponding weight coefficients a_i, to 15 decimal places.

The first such table, computed by Gauss gave 16 places up to $n=7$.[3] More recently work was done by Nyström,[4] who gave 7 decimals up to $n=10$, but for the interval $(-1/2, +1/2)$. B. de F. Bayly has given the roots and coefficients of $P_{12}(x)$ to 13 places.[5]

Figure: A comment about Gauss rules.

$n = 1$

$a_1 = 0,5$
$R_1 = 1$
Corr. $\frac{1}{12} L_2$

$n = 2$

$a_1 = 0,21132\ 48654$
$a_2 = 0,78867\ 51346$
$R_1 = R_2 = \frac{1}{2}$
Corr. $\frac{1}{180} L_4$

$n = 3$

$a_1 = 0,11270\ 16654$
$a_3 = 0,88729\ 83346$
$a_2 = 0,5$
$R_1 = R_3 = \frac{5}{18};\ R_2 = \frac{4}{9}$
Corr. $\frac{1}{2800} L_6$

$n = 4$

$a_1 = 0,06943\ 18442$
$a_4 = 0,93056\ 81558$
$a_2 = 0,33000\ 94782$
$a_3 = 0,66999\ 05218$
$R_1 = R_4 = 0,17392\ 74226$
$R_2 = R_3 = 0,32607\ 25774$
$\log R_1 = 9,24036\ 80612$
$\log R_2 = 9,51331\ 42764$
Corr. $\frac{1}{44100} L_8$

$n = 5$

$a_1 = 0,04691\ 00770$
$a_5 = 0,95308\ 99230$
$a_2 = 0,23076\ 53449$
$a_4 = 0,76923\ 46551$
$a_3 = 0,5$
$R_1 = R_5 = 0,11846\ 34425$
$R_2 = R_4 = 0,23931\ 43352$
$R_3 = \frac{64}{225} = 0,28444\ 44444$
$\log R_1 = 9,07358\ 43490$
$\log R_2 = 9,37896\ 87142$
$\log R_3 = 9,45399\ 74559$
Corr. $\frac{1}{698544} L_{10}$

$n = 6$

$a_1 = 0,03376\ 52429$
$a_6 = 0,96623\ 47571$
$a_2 = 0,16939\ 53068$
$a_5 = 0,83060\ 46932$
$a_3 = 0,38069\ 04070$
$a_4 = 0,61930\ 95930$
$R_1 = R_6 = 0,08566\ 22462$
$R_2 = R_5 = 0,18038\ 07865$
$R_3 = R_4 = 0,23395\ 69673$
$\log R_1 = 8,93278\ 94580$
$\log R_2 = 9,25619\ 02763$
$\log R_3 = 9,36913\ 59831$
Corr. $\frac{1}{11099088} L_{12}$

Kugel.

$n = 7$

$a_1 = 0,02544\ 60438\ 286202$
$a_7 = 0,97455\ 39561\ 713798$
$a_2 = 0,12923\ 44072\ 003028$
$a_6 = 0,87076\ 55927\ 996972$
$a_3 = 0,29707\ 74243\ 113015$
$a_5 = 0,70292\ 25756\ 886985$
$a_4 = 0,5$
$R_1 = R_7 = 0,06474\ 24830\ 844348$
$R_2 = R_6 = 0,13985\ 26957\ 446384$
$R_3 = R_5 = 0,19091\ 50252\ 525595$
$R_4 = \frac{256}{1225} = 0,20897\ 95918\ 367347$
$\log R_1 = 8,81118\ 93529$
$\log R_2 = 9,14567\ 08421$
$\log R_3 = 9,28084\ 01093$
$\log R_4 = 9,32010\ 38766$
Corr. $\frac{1}{176679360} L_{14}$.

Figure: Gaussian rules in $(0, 1)$ listed by Heine.

One aims to get similar rules for more general multivariate Ω as

- triangles,
- disks,
- squares,
- cubes,
- spheres,
- tori,

or even more general 2D and 3D regions.

The typical request is that these rules must have few nodes, high ADE (better if almost-minimal!), to be at hand very quickly, with small values of the conditioning of the $\text{cond}(\{w_i\})$.

See Approximation pointsets and quadrature rules for a list of rules on these domains.

We intend to investigate the case of polygons and polyhedra, in view of their interest in applications, e.g. the solution of PDE via FEM and VEM.

Paper

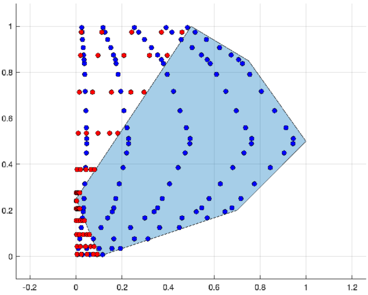
Product Gauss cubature over polygons based on Green's integration formula (2007)

- Purpose: cubature formula over convex, nonconvex Ω.
- ADE: the formula has algebraic degree of exactness ADE equal to $2n-1$, i.e. denoting by $\mathbb{P}_{2n-1}$ the space of polynomials of total degree $2n-1$,

$$\int_{\Omega} p(x,y)dxdy = \sum_{k=1}^{N_{2n-1}} w_k p(x_k, y_k), \text{ for all } p \in \mathbb{P}_{2n-1},$$

  with $N_{2n-1} \approx m\, n^2$, where "m" is the number of sides that are not orthogonal to a given line, and not lying on it.
- Preprocessing: it does not need any preprocessing like triangulation of the domain, but relies directly on univariate Gauss-Legendre quadrature via Green's integral formula.

Example: quadrature over polygons (meshfree approach)



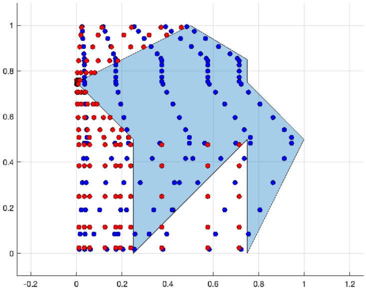


Figure: Examples of polygonal domains (ADE=9).

Left: a convex domain with 6 sides (180 nodes), Right: a non-convex domain with 9 sides (255 nodes).

Red dots: nodes with negative weights. Blue dots: nodes with positive weights.

- **Pros**: meshfree approach.
- **Cons**: In general these rules may have external nodes as well as negative weights.
- **Pros**: With some trick based on roto-translations we can compute rules over convex domains of PI-type (positive weights and internal nodes).
- **Cons**: Similarly, by rototanslations with we can compute rules with internal nodes even over some non-convex domains but in general this is not always possible.
- **Cons**: Rule may have high cardinality if the polygon has many sides.

We prefer rules with

- positive weights (more numerical stability and application to hyperinterpolation),
- internal nodes (integrand may not be defined outside the domain),

i.e. of PI-type.

Paper

Compressed cubature over polygons with applications to optical design (2020)

- Purpose: cubature formula over convex, nonconvex or even multiply connected polygons Ω.
- Strategy: once a minimal triangulation is available (see Matlab `polyshape` toolbox), we obtain the rule by applying an almost-minimal rule of PI-type on each triangle with the wanted ADE, summing the contributions.

Some remarks

- minimal triangulation: one can triangulate a general M-sides polygon via $M-2$ triangles (easy task in a convex polygon, not trivial for a general one),
- almost-minimal rule the number of its nodes is almost minimal between those having a certain ADE and requirement on the nodes (e.g. internal) and weights (e.g. positive).

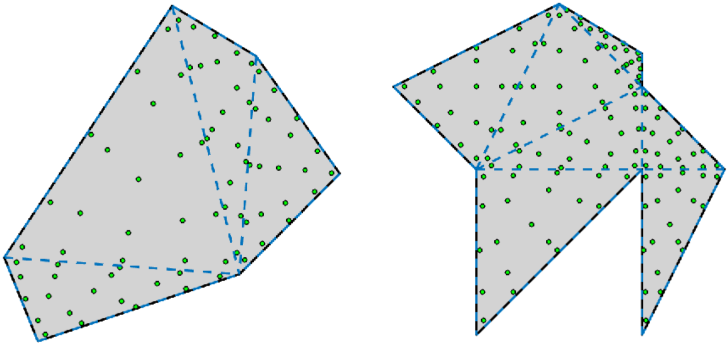


Figure: Examples of polygonal domains (ADE is $\delta = 9$).

Left: a convex domain with 6 sides (77 nodes, the previous rule had 180 nodes), Right: a non-convex domain with 9 sides (133 nodes, the previous rule had 255 nodes).

Note: all the weights are positive.

- **Cons**: requires triangulation.
- **Pros**: In general these rules always have internal nodes as well as positive weights.
- **Cons**: Rule still may have high cardinality if the polygon has many sides.

Observe that in the examples above ADE is $\delta = 9$

- convex domain: the rule has 77 nodes,
- not convex domain: the rule has 133 nodes.

Remark

In both cases the number of nodes is higher than the dimension of the polynomial space $\mathbb{P}_9$ that is equal to $(9+1)(9+2)/2 = 55$.

Our project is to quickly extract from the previous one, another rule of PI-type with the same degree of precision but with a number of nodes at most equal to dim$(\mathbb{P}_9)$ (i.e. a rule compression).

Tchakaloff theorem asserts that:

For every compactly supported measure there exists a positive algebraic quadrature formula of ADE δ with cardinality not exceeding the dimension ν_δ of $\mathbb{P}_\delta$ (restricted to the measure support)

In a variant it has been proved that

If PI-type cubature rule has

- *ADE equal to δ,*
- *cardinality $N_\delta > dim(\mathbb{P}_\delta(\Omega))$*

then we can extract one of PI-type with ADE equal to δ and cardinality at most $\nu_\delta = dim(\mathbb{P}_\delta(\Omega))$.

Paper

M. Putinar, A Note on Tchakaloff's Theorem, Proceedings of the American Mathematical Society Vol. 125, No. 8 (Aug., 1997), pp. 2409–2414.

Given

- a formula of PI-type with ADE equal to δ, nodes $\{\mathbf{x}_k\}_{k=1,\ldots,N_\delta}$ and weights $\{w_k\}_{k=1,\ldots,N_\delta}$,
- a basis $\{\phi_1,\ldots,\phi_{\nu_\delta}\}$ of $\mathbb{P}_\delta(\Omega)$,

let

- $V_{i,j} = (\phi_j(\mathbf{x}_i))$ the Vandermonde matrix at the nodes,
- $\boldsymbol{\gamma} = (\gamma_j)_{j=1,\ldots,\nu_\delta}$ where $\gamma_j = \int_\Omega \phi_j d\mu = \sum_{i=1}^{N_\delta} w_i\phi_j(\mathbf{x}_i)$, the vector of the moments (w.r.t. μ).

The problem mentioned above can be rephrased into computing a nonnegative solution with at most $\nu_\delta = \dim(\mathbb{P}_\delta(\Omega))$ nonvanishing components to the underdetermined linear system

$$V^T\mathbf{w} = \boldsymbol{\gamma}.$$

This can be accomplished by Lawson-Hanson active set method for NonNegative Least Squares (NNLS), implemented by the Matlab built-in routine `lsqnonneg` or in Python by `nnls`.

By this algorithm,

- adopting as basis $\{\psi_j\}$ the total-degree product Chebyshev basis of the smallest Cartesian rectangle $[a_1, b_1] \times [a_2, b_2]$ containing Ω, with the graded lexicographical ordering,
- from the PI-rules with ADE 9 obtained via triangulation,

we get the PI-rules below with cardinality $55 = \dim(\mathbb{P}_9)$.

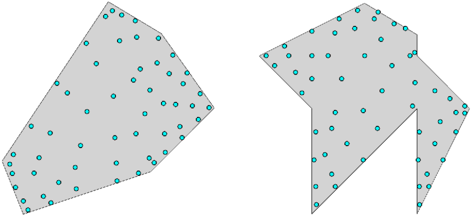


Figure: Examples of polygonal domains (ADE=9).
Left: a convex domain with 6 sides (55 nodes, the previous rule had 77 nodes), Right: a non-convex domain with 9 sides (55 nodes, the previous rule had 133 nodes).

Remark (When do not apply this technique)

Note that this approach is useful only when the initial rule of PI-type with ADE δ *has cardinality higher than the dimension* ν_δ *of* $\mathbb{P}_\delta(\Omega)$.

Thus it is worthless in the case of classical domains as the interval, the disk, simplex, cube, sphere, where there are explicit rules of PI-type with cardinality inferior to L.

Remark (Cputimes on the previous domains)

For mild ADE the computation of these compressed rules is fast. Running Matlab R2022a, on a computer with an Apple M1 processor and 16GB of RAM, we had average cputimes as in the table below:

Domain	tri. rule	compress.
Convex domain	*1.1-3s*	*1.7-2s*
Non-convex domain	*5.2-3s*	*1.1-2s*

Table: *Average cputime for computing rules with ADE=9 in the polygonal domains used in the tests.*

In the paper

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

the author proves a result that can be so summarized

a sufficiently dense pointset of the compact domain Ω contains the nodes of a PI-type rule with algebraic degree of precision ADE equal to δ.

Algorithm (sketch)

- *Moment computation: determine* $\{\gamma_k\}_{k=1,\dots,\nu_\delta}$ *of a certain polynomial basis* $\{\phi_k\}_{k=1,\dots,\nu_\delta}$ *of* $\mathbb{P}_\delta(\Omega)$*;*
- *Pointset: using an* `in-domain` *routine on a mesh in a domain* $\mathcal{R}$ *containing* Ω*, determine a sufficient number of points* $\{\tilde{P}_l\}_{l=1,\dots,N_\delta}$ *inside* Ω *so that the overdetermined linear system*

$$V^T w = \gamma, \quad V_{l,k} = (\phi_k(\tilde{P}_l))$$

  has a nonnegative solution w *with at most* $\dim(\mathbb{P}_\delta)$ *positive components.*
- *Rule extraction: solve* $V^T w = \gamma$ *via Lawson-Hanson algorithm. The non zero components of* w *determine the points and the weights of the wanted rule.*

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.

- Choice of the polynomial basis $\{\phi_k\}_{k=1,\dots,\nu_\delta}$ of $\mathbb{P}_m$ is not trivial. It must
  - provide Vandermonde matrices not too badly conditioned,
  - allow a fast computation of the moments

$$\gamma_k = \int_\Omega \phi_k(\mathbf{x})\, d\mu, \quad k = 1, \dots, \nu_\delta.$$

- Availability of the in-domain routine (e.g. Matlab `inpolygon`).
- Moment computation, e.g. by Gauss-Green approach in bivariate domains, or a divergence theorem based in trivariate domains;

This approach is meshless, after some not trivial efforts can be generalized to curvilinear polygons whose boundary is tracked parametrically by NURBS.

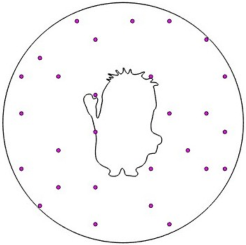


Figure: Example of NURBS type domain. A non-convex and not simply connected domain and the 28 nodes of a PI type rule with ADE=6.

Paper

A. Sommariva and M. Vianello, Low cardinality Positive Interior cubature on NURBS-shaped domains, BIT Numer. Math. 63, 22 (2023).

Summarizing, we saw 4 approaches to determine rule with ADE equal to δ.

- Meshfree NO/PI type rules: algorithm quickly provides rules that may have external points, negative weights and cardinality higher than the dimension of the polynomial space $\nu_\delta = (\delta+1)(\delta+2)/2$;
- triangulation based PI type rules: algorithm based on triangulation that rapidly provides rules that may have cardinality higher than the dimension of the polynomial space $\nu_\delta = (\delta+1)(\delta+2)/2$;
- PI type rules compression rules: algorithm based on a variant of Tchakaloff theorem that from PI rules with cardinality higher than $(n+1)(n+2)/2$ gives a PI rule with cardinality $\nu_\delta = (\delta+1)\delta+2)/2$;
- meshfree PI type rules: algorithm that provides PI rules with cardinality $\nu_\delta = (\delta+1)(\delta+2)/2$ (even on a large family curvilinear polygons).

We now 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 $\Omega \subset \mathbb{R}^3$.

- Lack for open-source routines in Matlab.
- The intention is to provide algorithms with and without tetrahedralization.
- *Mild* algebraic degree of exactness ADE *mild* (say less than 10).

This approach via tetrahedralization is well-known in literature.

- Determine a tetrahedralization $\mathcal{T} = \{T_k\}_{k=1,\ldots,M}$ 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 $Q_\delta^{(k)}(f) = \sum_{j=1}^{N_\delta} w_j^{(k)} f(P_j^{(k)})$ by a rule with algebraic degree of exactness δ on each T_k, $k = 1, \ldots, M$.
- In view of the additivity of the integration operator we get a rule of degree δ on Ω, i.e.

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

Some considerations about the **triangulation**.

- If the polyhedron Ω is not convex/star shaped (knowing a center!), the determination of the tetrahedralization may not be straightforward.
- If Ω is obtained by `alphashape` from a point cloud of vertices, the Matlab command `alphaTriangulation` returns a tetrahedralization of Ω. Similar routines are available in Python.
- 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 $ADE = 0, 1, \ldots, 20$, 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.

- All these rules have internal nodes and positive weights.
- For $ADE > 20$, 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 with ADE equal to δ is available on the reference tetrahedron T^*, it can be easily obtained on each tetrahedron T_k of the triangulation 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

$$\nu_\delta = dim(\mathbb{P}_\delta) = (\delta+1)(\delta+2)(\delta+3)/6$$

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.

Remark (B plan)

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

The procedure for determining a rule with ADE equal to δ works essentially as follows:

- we compute the moments $\{\gamma_k\}_{k=1,\dots,\nu_\delta}$ of a certain polynomial basis $\{\phi_k\}_{k=1,\dots,\nu_\delta}$ of tensorial type by means of cubature rules with ADE equal to $\delta + 1$ 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 $\nu_\delta \leq L$ 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.

For all these details see:

Paper

TetraFreeQ: tetrahedra-free quadrature on polyhedral elements (2024)

Remark (Software)

The Matlab routines are available as open-source codes as Approximation pointsets and quadrature rules

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

1. generate a *sufficiently dense* 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 routine `lsqnonneg` or Python procedure `nnls`,
   - extract a set of nodes $\mathcal{Q}^{(1)} = \{Q_j\}_{j=1}^{k_3} \subseteq \mathcal{P}^{(2)}$,
   - compute the relative (positive) weights $\{w_j\}_{j=1}^{k_3} \subseteq \mathbb{R}^+$,

Important.

Wilhelmsen's theorem says that in theory this procedure will have success for sufficiently dense data.

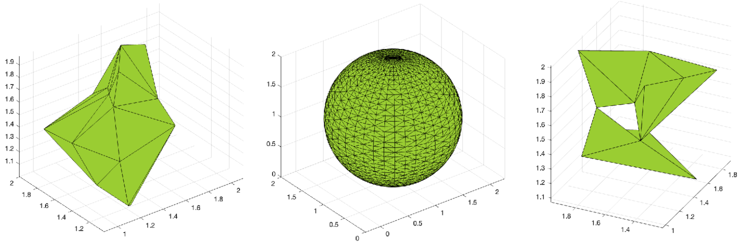


Figure: Examples of polyhedral domains.

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

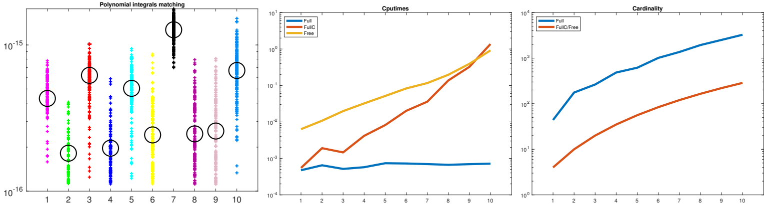


Figure: Domain 1 (30 facets):
Left: Integration errors over 100 polynomial integrands of the form $(c_1 + k_1 \cdot x + k_2 \cdot y + k_3 \cdot z)^\delta$ where $c_1, k_1, k_2, k_3 \in [0, 1]$ are random,
Center: average cputime;
Right: cardinality.

Triangulation cputime: $5e - 3$ seconds.

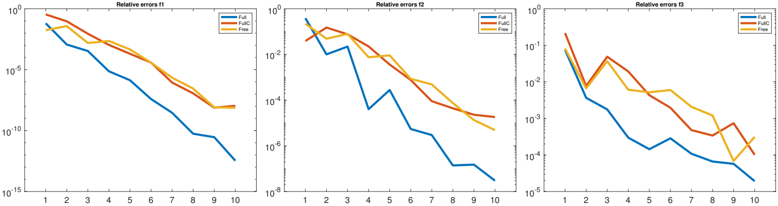


Figure: Domain 1 (30 facets): Relative errors integrating

- $f_1(x, y, z) = \exp(-x^2 - y^2 - z^2)$,
- $f_2(x, y, z) = ((x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2)^{5/2}$,
- $f_3(x, y, z) = ((x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2)^{1/2}$, where

$(x_0, y_0, z_0) = (1.5, 1.5, 1.5)$.

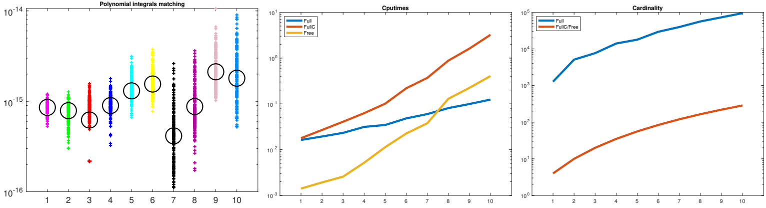


Figure: Domain 2 (760 facets, sphere like):
Left: Integration errors over 100 polynomial integrands of the form $(c_1 + k_1 \cdot x + k_2 \cdot y + k_3 \cdot z)^{\delta}$ where $c_1, k_1, k_2, k_3 \in [0, 1]$ are random,
Center: average cputime;
Right: cardinality.

Triangulation cputime: $8e - 2$ seconds. The indomain is fast since the domain is convex.

Numerical experiments: domain 2, integration of some functions

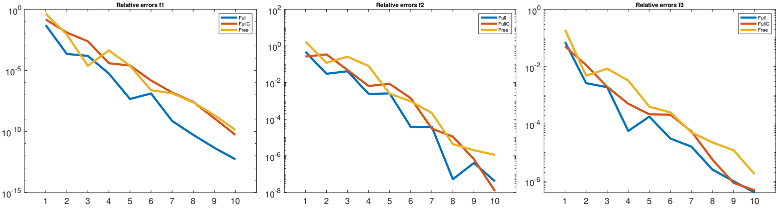


Figure: Domain 2 (760 facets, sphere like): Relative errors integrating

- $f_1(x, y, z) = \exp(-x^2 - y^2 - z^2)$,
- $f_2(x, y, z) = ((x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2)^{5/2}$,
- $f_3(x, y, z) = ((x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2)^{1/2}$, where

$(x_0, y_0, z_0) = (1.5, 1.5, 1.5)$.

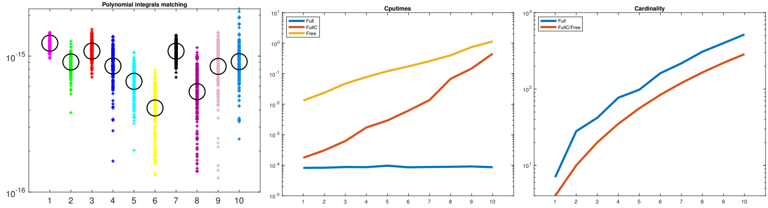


Figure: Domain 3 (20 facets, hole): Integration errors over 100 polynomial integrands of the form $(c_1 + k_1 \cdot x + k_2 \cdot y + k_3 \cdot z)^\delta$ where $c_1, k_1, k_2, k_3 \in [0, 1]$ are random,
Center: average cputime;
Right: cardinality.

Triangulation cputime: $5e-3$ seconds.

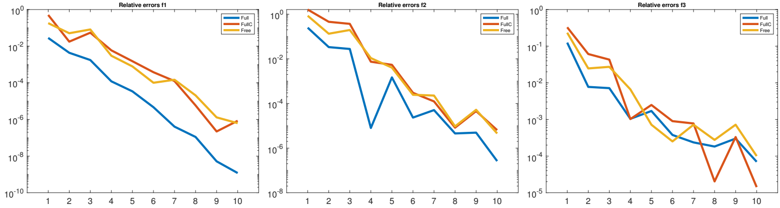


Figure: Domain 3 (20 facets, hole): • $f_1(x,y,z) = \exp(-x^2 - y^2 - z^2)$,
• $f_2(x,y,z) = ((x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2)^{5/2}$,
• $f_3(x,y,z) = ((x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2)^{1/2}$, where $(x_0, y_0, z_0) = (1.5, 1.5, 1.5)$.

Let

- Ω be a domain of $\mathbb{R}^2$ or $\mathbb{R}^3$,
- $f \in C(\Omega)$.

What we have done so far is to numerically approximate the integration functional

$$\mathcal{L}(f) := \int_{\Omega} f(x) d\Omega$$

by a discrete one

$$\mathcal{L}_{\delta}(f) := \sum_{i=1}^{N_{\delta}} w_i f(P_i)$$

so that the degree of exactness ADE is δ, i.e.

$$\mathcal{L}(p) = \mathcal{L}_{\delta}(p), \quad p \in \mathbb{P}_{\delta},$$

being $\mathbb{P}_{\delta}$ the set of polynomials over Ω, of total degree δ.

Suppose that

- $\Omega \subseteq B \subset \mathbb{R}^d$, where B is a parallelepiped (usually named *bounding box*),
- $f \in C(B)$;
- $\{\phi_k\}_{k=1,\ldots,\nu_\delta}$ is an orthonormal basis of $\mathbb{P}_\delta$, w.r.t. some absolutely continuous measure μ on B.

One can approximate f in B by the polynomial hyperinterpolant

$$\mathcal{H}_\delta(f) = \sum_{k=1}^{\nu_\delta} (f, \phi_k)\phi_k$$

where

$$(f, \phi_k) = \sum_{i=1}^{N_\delta} u_i f(P_i)\phi_k(P_i)$$

in which the r.h.s. is a cubature formula with positive weights u_i, $\{P_i\} \subset B$, with $ADE = 2\delta$.

Our approach to obtain such $\mathcal{L}_\delta$ is essentially the following:

- *compute a polynomial hyperinterpolant* $\mathcal{H}_\delta(f)$ *of* f *on a hypercube* B *containing the domain* Ω*;*
- *apply the integration functional* $\mathcal{L}$ *to the hyperinterpolant* $\mathcal{H}_\delta$*, i.e.*

$$\mathcal{L}(f) \approx \mathcal{L}(\mathcal{H}_\delta(f)) = \mathcal{L}_\delta(\mathcal{H}_\delta(f)) = \sum_{k=1}^{\nu_\delta} (f, \phi_k)\mathcal{L}_\delta(\phi_k).$$

that is

$$\int_\Omega f(x)d\Omega \approx \sum_{k=1}^{\nu_\delta} (f, \phi_k) \sum_{i=1}^{N_\delta} u_i f(P_i)\phi_k(P_i)$$

As in the similar case of classical interpolatory rules, these ideas can be converted in determining nodes and weights in the bounding box so that the rule has degree of exactness M on the integration domain Ω.

Suppose that the integration domain Ω is a polyhedron. Determine:

1. a Cartesian bounding box for the polyhedron;
2. the nodes $\{P_i\}$ and weights $\mathbf{u} = \{u_i\}$, $1 \leq i \leq N_\delta$, of a cubature formula exact for $\mathbb{P}_{2\delta}$ for a given absolutely continuous measure $d\mu = \sigma(P)dP$ on the bounding box;
3. an orthonormal basis $\{\phi_1, \ldots, \phi_\nu\}$ of $\mathbb{P}_\delta$ with respect to $d\mu$,
4. the corresponding moments $\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_\delta\}$ on Ω, $m_j = \int_\Omega \phi_j(P)\, dP$, e.g. by the divergence theorem;
5. the Vandermonde-like matrix

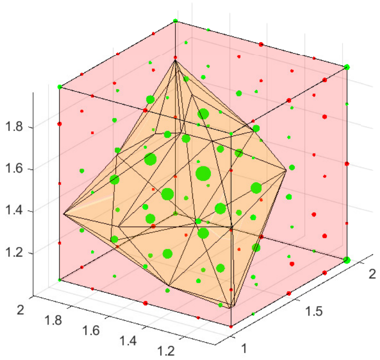
$$V = V_n(\{P_i\}) = [\phi_j(P_i)] \in \mathbb{R}^{N_\delta \times \nu}$$

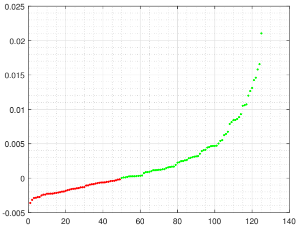
6. compute the weights as

$$\mathbf{w} = diag(\mathbf{u})\, V\boldsymbol{\gamma}\ ,$$

or in a Matlab-like notation $\mathbf{w} = \mathbf{u}. * \ V\boldsymbol{\gamma}$.

Example





- On the left: quadrature nodes of the cheap rule on the nonconvex polyhedral element Ω_1 for degree 4, as red dots if the pertinent weight is negative, as green dots otherwise. The size of the dots is visually proportional to the weight magnitude.
- On the right, distribution of the weights in increasing order (in red: negative weights, in green: positive weights). We report that the smaller weight is $w_{\text{min}} \approx -3.6 \cdot 10^{-3}$, the larger is $w_{\text{max}} \approx 2.1 \cdot 10^{-2}$, and the smaller size is $|w|_{\text{min}} \approx 2.6 \cdot 10^{-5}$.

Pros:

- Application to polytopal FEM (fast and stable computation of the integrals of products of polynomials naturally arising on arbitrary polyhedral elements, avoiding sub-tessellation into tetrahedra);
- moment computation does not require polyhedral tesselation;
- many computations can be done just once and repeated on different integration domains;
- w.r.t. similar recent techniques
  - cubature stability is ensured;
  - no QR factorization or linear system solution is involved.

Cons:

- Though stability is ensured some weights may be negative;
- the integrands require in general evaluations outside the integration domain.

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. S. Holcombe, *Matlab open source routine:* `inpolyhedron`
   Open source Matlab in-domain routine for polyhedra.
3. 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.
4. A. Sommariva, M. Vianello, Compression of multivariate discrete measures and applications, *Numer. Funct. Anal. Optim.*, 36 (2015), 1198–1223.
   (Details on cubature compression)
5. D. R. Wilhelmsen, A Nearest Point Algorithm for Convex Polyhedral Cones and Applications to Positive Linear approximation, *Math. Comp.*, 30 (1976), 48-57.