COMPRESSED CUBATURE OVER POLYGONAL DOMAINS

Version December 2 , 2018

B. BAUMAN *, A. SOMMARIVA †, AND M. VIANELLO ‡

Abstract. In this paper we propose an algorithm to determine cubature rules of algebraic degree of precision δ, on general polygons $\mathcal{P}$ by means of Matlab `polyshape` objects and almost minimal rules on triangles, finally obtaining by Lawson and Hanson algorithm, a formula with positive weights, nodes in $\mathcal{P}$ and cardinality at most $N_\delta = (\delta+1)(\delta+2)/2$. First, we test our algorithm on domains with different geometries and then on polygonal obscured and vignetted pupils, as required in some problems in optical astronomy.

Key words. Numerical cubature, polygonal domains, minimal triangulation, minimal formulae, Caratheodory-Tchakaloff theorem.

1. Introduction.

In this paper we propose how to compute cubature rules with algebraic degree of precision $ADE = \delta$ on polygons $\mathcal{P}$, that are in general not simple neither simply connected or connected.

These rules will have positive weights and nodes inside $\mathcal{P}$, i.e. of PI-type.

As general strategy we first triangulate the polygon using the Matlab `polyshape` routines, using a number of triangles that is minimal or almost minimal, and then use an almost minimal rule of PI-type on each triangle, with $ADE = \delta$, we determine the desired rule of PI-type on $\mathcal{P}$.

Similar techniques are not new, but this one has the advantage of treating very general polygons as well as of introducing cubature formulae with nodes $\{x_i\}$ whose cardinality M is in usually inferior than other general purpose codes.

Next, by solving a certain NNLS problem, by means of the Lawson-Hanson algorithm, we compute a rule of PI-type with nodes $\{t_i\} \subseteq \{x_i\}$ whose cardinality m is at most $N_\delta = (\delta+1)(\delta+2)/2$. Such set exists in view of the well-known Caratheodory-Tchakaloff theorem.

We compare these new sets, with previous ones on some polygons, convex, concave or even not simply connected, showing the advantages of our technique.

Finally, we test our approach on polygonal obscured and vignetted pupils, useful to solve certain problems in optical astronomy.

2. About the subdivision.

One of the key points to determine a rule with *few* cubature nodes over a polygon $\mathcal{P}$, consists in its partitioning in few simpler regions $\Omega_1, \ldots, \Omega_N$ and the application of a known formula on each of them.

In [33], the authors considered the case of *simple* polygons $\mathcal{P}$, i.e. without self intersections, having the so called *axis-property* for which there exists a *base-line* (say l), whose intersection with the polygon is connected, and such that in addition each line orthogonal to it (say q) has a connected intersection (if any) with the polygon $\mathcal{P}$, containing the point $l \cap q$. This class includes all convex polygons, for example by taking as l the line connecting a pair of vertices with maximal distance but also certain nonconvex polygons. Once that this baseline is at hand, without the need

*Lawrence Livermore National Laboratory, Livermore, United States, (`E-mail: bauman3@llnl.gov`)

†University of Padua, Department of Mathematics, via Trieste 63, Padua, Italy, (`E-mail: alvise@math.unipd.it`)

‡University of Padua, Department of Mathematics, via Trieste 63, Padua, Italy, (`E-mail: marcov@math.unipd.it`)

of particular *triangulators*, a formula with nodes in $\mathcal{P}$ and positive weights can be obtained via Gauss-Green theorem (with a well-established notation, we will name these rules of *PI-type*, where P stands for *positive weights* and I for *nodes in the domain*).

If such a baseline is not available, however a cubature rule can still be computed but without the warranty of being of PI-type (since the weights are still positive these formulae turn be of PO-type, i.e. some nodes are outside the given domain $\mathcal{P}$, though the weights are positive).

In order to enlarge the class of domains for which we can determine a rule of of PI-type, in [1] we implemented in Matlab a naive version of the so called *minimal triangulation* (see also [3] or [36] for a survey on the subject). In few words, a triangulation of a simply connected and simple polygon $\mathcal{P}$ with more than 3 sides, is based on the two ears theorem (see [25]), i.e on the fact that $\mathcal{P}$ has at least two *ears*, which are triangles with two sides being the edges of $\mathcal{P}$ and the third one completely in its interior. The algorithm first computes such an ear, say Ω_i, then removes it from $\mathcal{P}$, and repeats iteratively the procedure on $\mathcal{P}\backslash\Omega_i$ until only one triangle is left. The union of all the ears and of the final triangle provides a triangulation of the domain.

About the complexity of the algorithm, after several attempts by different scientists, it has been shown in [8] that the determination of such a triangulation on a simple polygon $\mathcal{P}$ with n vertices costs $O(n)$ operations.

We point out that this decomposition is minimal, in the sense that for a simple and connected polygon that is also simply connected and with n sides then $\mathcal{P} = \cup_{i=1}^{n-2}\Omega_i$, where Ω_i are triangles that only overlap on the sides, $n-2$ being the lower cardinality between all the possible triangulations.

Alternatively, in [15], it has been observed that any simple and simply connected polygon with n vertices can be partitioned into μ quadrangles (some of which possibly degenerating into triangles) with $(n-2)/2 \leq \mu \leq n-2$, where μ is often close to the lower bound. For example, any convex polygon is trivially partitioned into $\mu = (n-2)/2$ quadrangles for μ even, or into $(\mu-3)/2$ quadrangles plus one triangle for μ odd, simply by taking quadruples of consecutive vertices. Then by rules that are almost minimal and of PI type on the simplex or the square or alternatively of tensorial type, one can easily achieve rules on each triangle or quadrilateral subdomain of the partition, and more generally a rule of PI-type on $\mathcal{P}$.

Recently Matlab introduced the `polyshape` environment, that manages a polygon defined by 2-D vertices via a polyshape object with properties describing its vertices, solid regions, and holes. This new class includes several facilities. For instance, it allows boolean operations, as intersection, difference, union, and symmetrical difference between polygons. Furthermore it operates rotation, scaling and translation of the given sets.

In this framework, one can triangulate each polygon $\mathcal{P}$, that can be of very general nature, even not simply connected, not simple or disconnected. We stress that Matlab does not provide any reference about this partitioning. We tested the quality of the triangulation $\{\Omega_i\}$ on very different polygonal domains $\mathcal{P}$ with n sides, achieving $\mathcal{P} = \cup_{k=1}^{N}\Omega_i$ with $N \sim n$ and very often $N = n-2$, as for the minimal triangulation in the case of simple, as well as simply connected polygons. The procedure turns out to be rather fast. In order to give a glimpse of the performance, we considered several polygons with a number of vertices n ranging from 100 to 20000.

As regions, we have taken

1. a regular polygon $\mathcal{P}^{(1)}$ whose n vertices are

$$P_k = (\cos(t_k), \sin(t_k))$$

where $t_k = \frac{2\pi k}{n}$, with $k = 1, \ldots, n$;
2. a polygonal cardioid $\mathcal{P}^{(2)}$ whose n vertices are

$$P_k = (\cos(t_k) \cdot (1 - \cos(t_k)), \sin(t_k) \cdot (1 - \cos(t_k)))$$

where $t_k = \frac{2\pi k}{n}$, with $k = 1, \ldots, n$;
3. a polygonal Bernoulli lemniscate $\mathcal{P}^{(3)}$ whose n vertices are

$$P_k = (\sqrt{2}\cos(t_k)/(1 + \sin^2(t_k), \sqrt{2}\cos(t_k)\sin(t_k)/(1 + \sin^2(t_k)),$$

where $t_k = \frac{2\pi k}{n}$, with $k = 1, \ldots, n$.

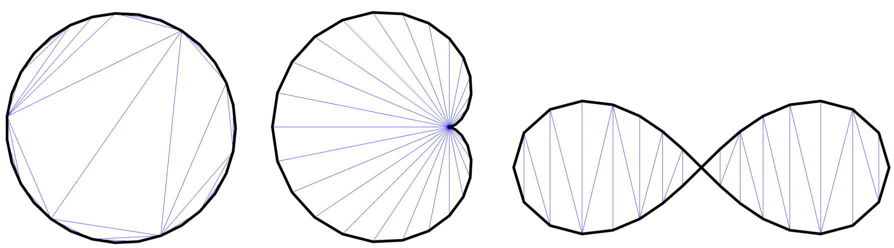

FIG. 2.1. *From left to the right, the three polygons $\mathcal{P}^{(1)}$, $\mathcal{P}^{(2}$, $\mathcal{P}^{(3)}$ for $n = 32$, and their triangulation.*

Observe that the polygonal Bernoulli Lemniscate is a non simple domain, due to the self intersection in the origin, but can be still correctly treated until $n = 1000$. From $n = 2000$ on, the number of triangles becomes inferior to $n - 4$ that corresponds to the cardinality of the minimal triangulation.

We made 10 tests for each subcase, computing in each one first a polyshape object and then a triangulation. The average cputimes for each polygon $\mathcal{P}^{(k)}$, $k = 1, \ldots, 3$ are listed in Table 2.1, respectively as t_P and t_T, an asterisk meaning that the triangulation process presents warnings.

TABLE 2.1
Cputime of triangulation of several polygonal domains. Asterisk means that the triangulation process presents warnings.

vertices	$\mathcal{P}^{(1)}$		$\mathcal{P}^{(2)}$		$\mathcal{P}^{(3)}$	
	t_P	t_T	t_P	t_T	t_P	t_T
100	$4e-03$	$2e-03$	$3e-03$	$1e-03$	$3e-03$	$1e-03$
500	$4e-03$	$3e-03$	$4e-03$	$4e-03$	$4e-03$	$6e-03$
1000	$1e-02$	$6e-03$	$1e-02$	$1e-02$	$1e-02$	$2e-02$
2000	$4e-02$	$7e-03$	$5e-02^*$	$4e-02^*$	$2e-02^*$	$5e-02^*$
3000	$8e-02$	$1e-02$	$8e-02^*$	$6e-02^*$	$4e-02^*$	$1e-01^*$
4000	$1e-01$	$1e-02$	$2e-01^*$	$9e-02^*$	$7e-02^*$	$2e-01^*$
5000	$2e-01$	$2e-02$	$2e-01^*$	$1e-01^*$	$1e-01^*$	$3e-01^*$
10000	$8e-01$	$4e-02$	$1e+00^*$	$5e-01^*$	$5e-01^*$	$1e+00^*$
20000	$4e+00$	$6e-02$	$4e+00^*$	$1e+00^*$	$2e+00^*$	$3e+00^*$

All tests have been performed on a 2,7 GHz Intel Core i5 with 16 GB 1867 MHz DDR3 memory.

The results say that for polygons with less than $n \leq 3000$ vertices the triangulation cputime is negligible, while for $n > 3000$ the time t_P needed for the polyshape construction can be even dominant w.r.t. t_T.

3. On the cubature nodes in the subdivision. Once a subdivision $\mathcal{P} = \cup_{i=1}^{n} \Omega_i$ is at hand, in order to have a cubature rule of PI-type on $\mathcal{P}$, by the additivity of the integral operator, it is sufficient to define a rule of PI-type on each subdomain Ω_i, that usually are triangles or quadrilaterals.

In the first instance, relevant in this work, knowing a formula on a reference triangle $\mathcal{T}^*$, after the conversion of the nodes in barycentric coordinates, it is straightforward to achieve one on each possible triangle $\mathcal{T}$ by varying the weights proportionally to their area. In other words, suppose that $\mathcal{T}^*$ is the unit-triangle with vertices $(0,0)$, $(0,1)$, $(1,0)$, $\{\phi_k\}_{k=1,\ldots,N_\delta}$ is a basis of the space of polynomials $\mathbb{P}_\delta$ of total degree δ and cardinality $N_\delta = (\delta+1)(\delta+2)/2$, e.g. the Proriol-Dubiner basis [6], and that

$$\sum_{i=1}^{m} w_i^* \phi_k(\xi_i^*) = \int_{\mathcal{T}^*} \phi_k d\mathcal{T}^*, \;\; \xi_i \in \mathcal{T}^*, \;\; k = 1, \ldots, N_\delta. \tag{3.1}$$

has algebraic degree of exactness $ADE = \delta$, i.e. able to integrate exactly each algebraic polynomial of total degree δ over the region $\mathcal{T}^*$.

Furthermore let $\xi_i^* = (x_i^*, y_i^*)$ and denote by $\hat{\xi_i^*} = (x_i^*, y_i^*, 1-x_i^*-y_i^*)$ the barycentric coordinates of ξ_i^*, by $\mu(\mathcal{T})$ the area of $\mathcal{T}$, and by V_1, V_2, V_3 its vertices. Being $\mu(\mathcal{T}^*) = 1/2$, the cubature rule on $\mathcal{T}$ with nodes

$$\tilde{\xi}_i = x_i^* \cdot V_1 + y_i^* \cdot V_2 + (1 - x_i^* - y_i^*) \cdot V_3, \quad i = 1, \ldots, m$$

and weights

$$w_i = \frac{\mu(\mathcal{T})}{\mu(\mathcal{T}^*)} w_i^* = 2\mu(\mathcal{T}) w_i^*, \quad i = 1, \ldots, m$$

has $ADE = \delta$ on $\mathcal{T}$, i.e. integrates exactly each polynomial in $\mathbb{P}_\delta$ on $\mathcal{T}$.

In view of this well-known result, it is necessary to compute formulae with a certain ADE only on the reference triangle $\mathcal{T}^*$.

In this context, the more appealing ones to achieve a low cardinality rule are the so called *minimal rules* that have the lowest number of nodes between those whose algebraic degree of exactness ADE is δ. Differently from the univariate case, where they are the so called *Gaussian rules*, they are known only in few cases (see, e.g. [9]).

In spite of this, the bibliography about rules with low cardinality $\mathcal{T}$, with a certain algebraic degree of exactness (i.e. *almost minimal*, since in general it is quite difficult to prove that they are minimal), is rather wide. A common technique for their determination is the following. If $\{\phi_k\}_{k=1,\ldots,N_\delta}$ is a polynomial basis on Ω^* and all the moments $\gamma_k = \int_{\Omega^*} \phi_k d\Omega^*$, $k = 1, \ldots, N_\delta$ are known, one computes via numerical optimisation the solutions (ξ_k, w_k), with $k = 1, \ldots, M$ of the problem (3.1) with $M \leq N_\delta$ as low as possible. This task is not an easy one, especially when the number of equations N_δ becomes large (see, e.g., [37]). In order to lower the number of equations, symmetries of the nodes allow to search the optimal set in a less general family, as well as to solve smaller nonlinear systems. Variants of this successful strategy provided many almost minimal pointsets, even for rather large δ (see, e.g. [12], [23], [26], [27], [41], [43]).

In Table 3.1 we have listed to the authors' knowledge the best of these almost minimal rules between those of PI-type. that are used to implement a formula with algebraic degree of precision $ADE = \delta$, not only on $\mathcal{T}^*$ but also any $\mathcal{T}$ and more generally, by the additivity of the integral operator, on any polygon $\mathcal{P}$.

TABLE 3.1
Cardinality N^*_δ of (almost) minimal rules on triangles with $ADE = \delta$.

δ	N^*_δ	δ	N^*_δ	δ	N^*_δ	δ	N^*_δ	δ	N^*_δ
1	1	11	27	21	85	31	181	41	309
2	3	12	32	22	93	32	193	42	324
3	4	13	36	23	100	33	204	43	339
4	6	14	42	24	109	34	214	44	354
5	7	15	46	25	117	35	228	45	370
6	11	16	52	26	130	36	243	46	385
7	12	17	57	27	141	37	252	47	399
8	16	18	66	28	150	38	267	48	423
9	19	19	70	29	159	39	282	49	435
10	24	20	78	30	171	40	295	50	453

All the rules are stored as a Matlab file at [32], where we have corrected the formulae whenever necessary in order to match (3.1) close to machine precision.

For higher degrees one can use the well-known Stroud conical rules [22], whose cardinality is $(\delta+1)^2/4$ for odd δ.

Note: We point out that though it is not the purpose of the paper, many authors determined adaptive algorithms for numerical integration of functions defined on triangles (see, e.g. [4], [11], [17], [19], [18], [20], [40] and references therein). In particular one can take advantage of the vectorizing the procedure in Matlab as explained in [29], for integration on the square to implement a variant for adaptive cubature over polygons. It is our intention to investigate later this subject.

Note: Observe that one can use any triangulation of $\mathcal{P}$ to determine a formula of PI-type on such domain, but the minimal one has the advantage of keeping as low as possible the cardinality of the cubature formula, since in any triangle a fixed number of points is used.

4. Caratheodory-Tchakaloff subsampling. The purpose of this section is to show how from a rule with high cardinality, positive weights, and $ADE = \delta$ we can extract another with the same degree of precision, positive weights, but with cardinality equal at most to the dimension N_δ of the vector space $\mathbb{P}_\delta$.

In some recent papers, the authors have applied a mathematical tool named CATCH (acronym of (Caratheodory-Tchakaloff) subsampling), for such compression of discrete measures, proposing its application to discrete polynomial Least squares by sparse moment matching. In this framework, the method selects from a large discretization of a given region a much smaller number of (weighted) sampling points, even on a complex shape as can be the polygons investigated in this paper, keeping numerically invariant the Least Squares approximation estimates.

The key point is the following discrete version of Tchakaloff theorem [38] proved by Caratheodory theorem on finite-dimensional conic/convex combinations [7], [28],

THEOREM 4.1. *Let μ be a multivariate measure whose support is a $\mathbb{P}_\delta$-determining finite set $X = \{\xi_i\} \subset \mathbb{R}^d$ (i.e., n-degree polynomials vanishing there vanish everywhere), with correspondent positive weights (masses) $\lambda = \{\lambda_i\}$, $i = 1, \ldots, M$, $M = card(X) > N_\delta$, N_δ being the dimension of $\mathbb{P}_\delta$. Then, there exist a cubature formula for the discrete measure μ, with nodes $T_n = \{t_j\} \subset X$ and positive weights*

$\mathbf{w} = \{w_i\}$, $1 \leq j \leq m$, *with* $m \leq N_\delta$, *such that*

$$\int_X p(x)d\mu = \sum_{i=1}^{M} \lambda_i p(x_i) = \sum_{j=1}^{m} w_j p(t_j),\ \forall p \in \mathbb{P}_\delta.$$

From the numerical side, given any polynomial basis $\{\phi_k\}$ of $\mathbb{P}_\delta$, define the Vandermonde-like matrix $V = \{V_n(X)\}_{i,j} = \phi_j(\xi_i)$ and let γ the vector of moments of the polynomial basis $\{\phi_j\}$ with respect to the original discrete measure and consider the underdetermined moment system $V^T u = \gamma$.

Caratheodory/Tchakaloff theorem 4.1 asserts that there exists a sparse nonnegative solution u to the system above, whose nonvanishing components (i.e., the weights $\{w_j\}$) are at most N_δ and determines the corresponding reduced sampling points $T_\delta = \{t_j\} \subseteq X$, that we may term the Caratheodory-Tchakaloff (CATCH) points of X.

The computation of these nodes has been considered in several papers (see e.g. [34], [39]). 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

$$\begin{cases} \min_{u \geq 0} c^T u \\ V^T u = b, \qquad u \geq 0 \end{cases} \tag{4.1}$$

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 [28], [39].

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

$$\text{compute } u^* : \|Au^* - \gamma\|_2 = \min_u \|Au - \gamma\|_2, u \geq 0, \tag{4.2}$$

in which u^* can be obtained by the well-known *Lawson-Hanson* active set optimization method [21], which determines a sparse solution to (4.2). Its application gives a residual $\epsilon = \|Au^* - \gamma\|_2$ that is typically very small, say $< 10^{-14}$ for $\delta \leq 30$.

Our numerical experience with now available Matlab software has shown that NNLS usually performs better than LP in computing the CATCH weights, at least for moderate degrees δ (namely, N_δ for mild degrees) [28]. Consequently all our codes are based on this the application of Lawson-Hanson method to compute the Caratheodory/Tchakaloff sets.

We point out that there are several versions of NNLS codes available in Matlab. One is the built-in function `lsqnonneg`, based on the Lawson-Hanson algorithm while an open-source version present in the package NNLSlab in [31]. Other alternatives are often obtained by MEX files and for generality purposed will not be used here.

Note. As alternative, one can compress the rule via the QR algorithm with column pivoting proposed in 1965 by Businger and Golub [5], which is implemented for example by the Matlab *backslash* operator. This approach is equivalent to a greedy selection of the columns of A in order to maximize the successive volumes, and eventually the (absolute value of the) determinant (the column selection problem being NP-hard). If the matrix has full rank, the final result is a weight vector w^* where only $N \leq N_\delta = (\delta+1)(\delta+2)/2$ components are nonzero, so that we can extract the cubature nodes $T_\delta = \{t_j\}$ from X by the column indexes corresponding to such components. A drawback is that the resulting weights $w^* = \{w_k^*\}$ may not be all positive, but typically the negative ones are few and of small size, so that the relevant stability parameter

$$\sigma = \frac{\sum_{k=1}^{N} |w_k^*|}{|\sum_{k=1}^{N} w_k^*|} \geq 1$$

is not far from 1 (see also [34]).

Note. In the cubature framework an algorithm termed *Recursive Halving Forest*, based on a hierarchical SVD, has been proposed in [39]. Performances are reported for large scale problems (say that the order of N_δ is 10^3, 10^4). Unfortunately the software is not available and thus cannot be applied here as comparison.

5. Numerical experiments. As for the numerical experiments, we intend first to compare the results that we can obtain via this new algorithm with those in [33] and next to show the effect of Caratheodory-Tchakaloff subsampling in the compression of a cubature rule. Finally we propose a method to obtain embedded rules on complicated polygonal domains.

In [33], the numerical experiments where made respectively on a convex and on certain non-convex polygon, say $\Omega^{(C)}$, $\Omega^{(NC)}$, achieving rules of PI-type (see 5.1 for the description of the polygons, baselines and pointset distribution for $\delta = 10$).

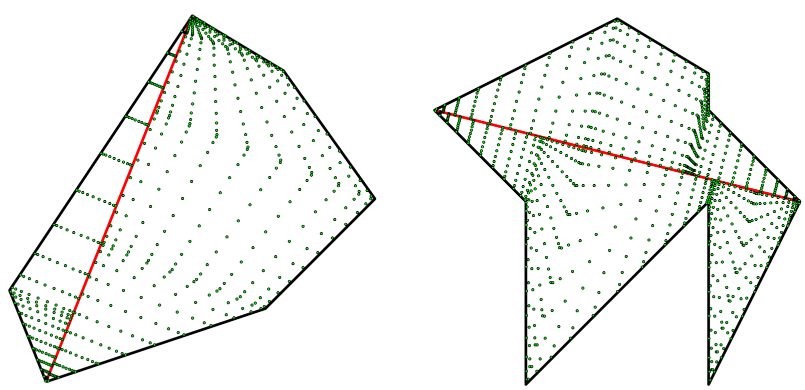

FIG. 5.1. *From left to the right, the domains $\Omega^{(C)}$, $\Omega^{(NC)}$. The red line is the axis used by the algorithm in [33]. The pointset is relative to $ADE = \delta = 10$.*

In Table 5.1, we compare the two approaches, as well as the cardinality of each cubature pointset, with cardinalities $M_\delta^{(old)}$, $M_\delta^{(new)}$, $\tilde{M}_\delta^{(new)}$ respectively applying the algorithm proposed in [33], the one presented here before and after the compression.

From this table, it is evident compression via NNLS provides a cubature pointset with cardinality $(\delta+1)(\delta+2)/2$, i.e. the dimension of the vector space $\mathbb{P}_\delta$ and that the compression ratio $M_\delta^{(new)}/\tilde{M}_\delta^{(new)}$ depends on the number of sides of the polygon.

As for the cputime, the computation of any of these rules is not an issue since their determination takes less than 10^{-2} seconds. Compression using Matlab built-in routine `lsqnonneg` or the open-source NNLSlab is in general not too time consuming for low degrees, while it becomes relevant when $\delta \geq 20$.

TABLE 5.1

Cardinality $M_\delta^{(old)}$, $M_\delta^{(new)}$, $\tilde{M}_\delta^{(new)}$, respectively applying the algorithm proposed in [33], the one presented here before and after the compression, on two polygons $\Omega^{(C)}$, $\Omega^{(NC)}$, with $ADE = \delta = 5, 10, \ldots, 40$.

	$\Omega^{(C)}$			$\Omega^{(NC)}$		
δ	$M_\delta^{(old)}$	$M_\delta^{(new)}$	$\tilde{M}_\delta^{(new)}$	$M_\delta^{(old)}$	$M_\delta^{(new)}$	$\tilde{M}_\delta^{(new)}$
5	180	28	21	235	49	21
10	660	96	66	870	168	66
15	1440	184	136	1905	322	136
20	2520	312	231	3340	546	231
25	3900	468	351	5175	819	351
30	5580	684	496	7410	1197	496
35	7560	912	666	10045	1596	666
40	9840	1180	861	13080	2065	861

In Table 5.2, we approximate the moments $\{\gamma_k\}$ of the basis $\{\phi_k\}$ consisting of the product of scaled Chebyshev polynomials on the smallest rectangle with sides parallel to the cartesian axes, containing the domain in analysis and whose total degree is inferior or equal to δ. In particular, in such polynomial basis, first we evaluate the moments $\{\gamma_k^{(T)}\}$ by the rule via triangulation and then compare them with those of the formulae obtained by compression (say $\{\tilde{\gamma}_k^{(T)}\}$) and Gauss-Green approach (say $\{\gamma_k^{(GG)}\}$), reporting $E_{T,TC} := \|\gamma_k^{(T)} - \tilde{\gamma}_k^{(T)}\|_2$, $E_{T,GG} = \|\gamma_k^{(T)} - \gamma_k^{(GG)}\|_2$.

For the compression we used in particular the software available from the package NNLSlab in [31].

TABLE 5.2

Comparison of the moments of cubature rules obtained resp. via triangulation (with/without compression) and Gauss-Green approach on two polygons $\Omega^{(C)}$, $\Omega^{(NC)}$, with $ADE = \delta$.

	$\Omega^{(C)}$		$\Omega^{(NC)}$	
δ	$E_{T,TC}$	$E_{T,GG}$	$E_{T,TC}$	$E_{T,GG}$
5	$3e-16$	$3e-16$	$2e-16$	$2e-16$
10	$3e-16$	$3e-16$	$3e-16$	$3e-16$
15	$4e-16$	$4e-16$	$5e-16$	$5e-16$
20	$8e-16$	$8e-16$	$6e-16$	$6e-16$
25	$6e-16$	$6e-16$	$1e-15$	$1e-15$
30	$2e-15$	$2e-15$	$1e-15$	$1e-15$
35	$1e-15$	$1e-15$	$2e-15$	$2e-15$
40	$2e-15$	$2e-15$	$2e-15$	$2e-15$

The estimates in Table 5.2 are relevant, since in [34] we proved that if $f \in C(\Omega)$ and λ is a measure over Ω (e.g. the scaled Chebyshev measure mentioned above or even a fully discrete one), ν the discrete measure such that $\int_\Omega f(P) d\nu = \sum_k w_k f(P_k)$, $\mathbb{P}_\delta = \mathrm{span}(\{\phi_k\})$, then denoting by $E_\delta(f) := \min_{p \in \mathbb{P}_\delta} \|f - p\|_\infty$, $\epsilon_{mom} := \|\mathbf{m} - \mathbf{m}^*\|_2$, where $\mathbf{m}_k := \int_\Omega \phi_k(P) d\nu$, $\mathbf{m}^*_{\mathbf{k}} := \sum_j w_j \phi_k(Q_j)$ we have

$$\left| \int_\Omega f(P) d\nu - \sum_k w_k f(Q_k) \right| \leq C E_\delta(f) + \|f\|_{L^2_{\lambda(\Omega)}} \epsilon_{mom}, \ \forall f \in C(\Omega)$$

in which

$$C \leq 2(\nu(\Omega) + \sqrt{\lambda(\Omega)} \epsilon_{mom})$$

if the cubature rule has nonnegative weights, as in our instances.

As consequence, we can expect that the behaviours of the three rules mentioned above is numerically *equivalent* when integrating a function $f \in C(\Omega)$, since all the quantities ϵ_{mom} are very small.

To show the advantages of this new approach via polyshape triangulation, we consider as examples two polygons $\Omega^{(QF_1)}$, $\Omega^{(QF_2)}$, that we can think as discretization of a quatrefoil, that could not be treated by the previous algorithms since the domains are not simple and have not the axis property.

More precisely,

$$\Omega^{(QF_i)} = (\cos(t_k) \cdot \sin(2t_k), \sin(t_k) \cdot \sin(2t_k)))$$

where $t_k = \frac{2k\pi}{M}$, with $k = 1, \ldots, M$. In particular, we set $M = 129$, and $M = 513$ respectively for $\Omega^{(QF_1)}$, $\Omega^{(QF_2)}$. The polygon $\Omega^{(QF_1)}$ can be partioned in $N_{tri}^{(QF_1)} = 120$ triangles, while $\Omega^{(QF_2)}$ in $N_{tri}^{(QF_2)} = 504$ triangles.

TABLE 5.3

Comparison of the moments of cubature rules obtained via triangulation (with/without compression) as well as their cardinalities $M_\delta^{(new)}$, $\tilde{M}_\delta^{(new)}$, on two not simple polygons $\Omega^{(QF_1)}$, $\Omega^{(QF_2)}$, with $ADE = \delta = 5, 10, \ldots, 35$.

	$\Omega^{(QF_1)}$				$\Omega^{(QF_2)}$			
δ	$E_{T,TC}$	$M_\delta^{(new)}$	$\tilde{M}_\delta^{(new)}$	Ratio$_\delta$	$E_{T,TC}$	$M_\delta^{(new)}$	$\tilde{M}_\delta^{(new)}$	Ratio$_\delta$
5	$1e-15$	840	21	40.0	$1e-15$	3528	21	168.0
10	$1e-15$	2880	66	43.6	$2e-15$	12096	66	183.3
15	$2e-15$	5520	136	40.6	$2e-15$	23184	136	170.5
20	$3e-15$	9360	231	40.5	$4e-15$	39312	231	170.2
25	$1e-14$	14040	351	40.0	$6e-15$	58968	351	168.0
30	$9e-15$	20520	496	41.4	$1e-14$	86184	496	173.8
35	$9e-15$	27360	666	41.1	$1e-14$	114912	666	172.5

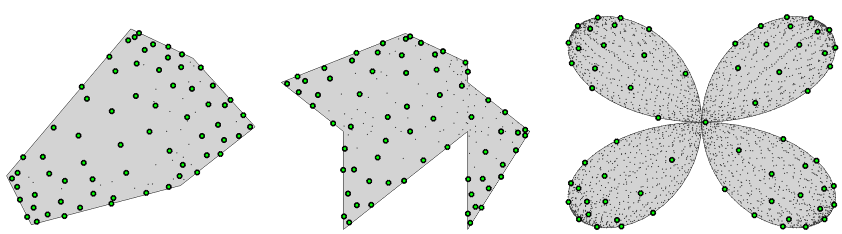

FIG. 5.2. *Compressed set at degree $\delta = 10$, for the polygons $\Omega^{(C)}$, $\Omega^{(NC)}$, $\Omega^{(QF_1)}$.*

By a careful look at Table 5.3, we observe that the compression ratio

$$\text{Ratio}_\delta = M_\delta^{(new)} / \tilde{M}_\delta^{(new)}$$

is almost constant. The reason is that at degree δ the cubature rule has approximatively $\delta^2/6$ points for triangle, thus $M_\delta^{(new)} \approx N_{tri}^{(QF_i)} \delta^2/6$ and since the compressed set has cardinality $\tilde{M}_\delta^{(new)} \approx \delta^2/2$, necessarily

$$\text{Ratio}_\delta = \frac{M_\delta^{(new)}}{\tilde{M}_\delta^{(new)}} \approx \frac{N_{tri}^{(QF_i)} \delta^2/6}{\delta^2/2} = \frac{N_{tri}^{(QF_i)}}{3}.$$

At a first glance the value $E_{T,TC}$ seems to deteriorate increasing δ, but one has to consider that the dimension $N_\delta = (\delta+1)(\delta+2)/2$ of the vector of the moments increases, so the *relative* norm two $E_{T,TC}/\sqrt{N_\delta}$ is not too far from to machine precision.

We finally observe that by Caratheodory-Tchakaloff subsampling we can determine embedded rules over a polygon $\mathcal{P}$, i.e. cubature formulae $Q^{(L)}$, $Q^{(H)}$,

$$Q^{(L)}(f) = \sum_{k=1}^{M_L} w_k^{(L)} f(P_k^{(L)}),\ Q^{(H)}(f) = \sum_{k=1}^{M_H} w_k^{(H)} f(P_k^{(H)})$$

with different degree of precision, say $\delta_{Q^{(L)}} < \delta_{Q^{(H)}}$, where $\{P_k^{(L)}\} \subseteq \{P_k^{(H)}\}$ (see also [35]).

A typical application is that they allow stopping criteria, based on rules with different degrees of precision δ_1, δ_2, with $\delta_1 < \delta_2$, with the property that the the nodes of the rule with $ADE = \delta_1$ are between those of the rule with $ADE = \delta_2$. Their purpose is to minimize the number of function evaluations, for providing an estimate of the cubature error at degree δ_1.

More generally, given a sequence of degrees $\delta_1 < \delta_2 < \cdots < \delta_k$, we can compute the nested CATCH sequence $\{T_{\delta_j}\}$

$$\mathcal{P} \supset T_{\delta_k} \supset T_{\delta_{k-1}} \supset \cdots \supset T_{\delta_2} \supset T_{\delta_1} \tag{5.1}$$

together with the corresponding sequence of positive weight vectors, say $\{\mathbf{w}_{\delta_j}\}$, by solving backward the sequence of NLLS problems

$$\text{compute } \mathbf{u}_{j-1}^* \ :\ \|A_j \mathbf{u}_{j-1}^* - \mathbf{b}_j\|_2 = \min \|A_j \mathbf{u}_{j-1} - \mathbf{b}_j\|_2\ ,\ \ \mathbf{u}_{j-1} \geq \mathbf{0}\ , \tag{5.2}$$

for $j = k+1, k, \ldots, 2$, where $A_j = (V_{\delta_{j-1}}(T_{\delta_j}))^t$, $\mathbf{b}_j = A_j \mathbf{u}_j^*$, and we set $T_{\delta_{k+1}} = X$, $\mathbf{u}_{k+1}^* = \lambda$.

6. Application to polygonal obscured and vignetted pupils. As application of these new cubature formulae, we consider a problem arising in optical design.

In [14], Forbes suggested to use product Gaussian quadrature over bivariate domains Ω that were filled, circular or elliptical apertures (pupils), to compute root-mean-square (rms) spot size for an optical design.

Later, Bauman and Xiao [2] studied cubature methods based on prolate spheroidal wave functions [42], to treat situations where the pupil is obscured and vignetted (a feature that occurs, for example, in optical astronomy; see figure 6.1). Indeed, Large Synoptic Survey Telescope (LSST) has a large central obscuration (about 62 per cent obscuration by diameter) as well as a considerable vignetting (of up to 10 per cent by area) making the techniques developed by Forbes not appropriate for the problem.

In [10] the authors took into account an example of circular pupil (the unit disk) which is obscured by a central smaller disk and clipped by a circular arc of larger radius, similar to that appearing in [2], providing algebraic rules of PI-type with $ADE = \delta$.

In all those examples the pupils where circular, but in real applications they can also have a different geometry, e.g. regular polygons with many sides. To this purpose, we introduce here a novel approach to determine rules of PI-type having $ADE = \delta$ on domains that are based on *polygonal* pupils.

As example, we take into account systems where the polygon $\mathcal{P}$ is the set subtraction of two polygonal regions $\mathcal{P}^{(O)}$, $\mathcal{P}^{(I)}$ where

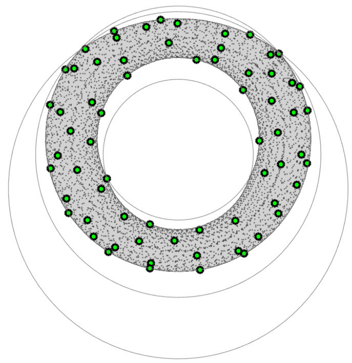

FIG. 6.1. *We have drawn the circles determining the domain $\Omega^{(M_1)}$, i.e. a polygonal obscured and vignetted pupil, the compressed pointset for $\delta = 10$ is determined by the green circles, while the initial one by small black dots.*

1. the polygon $\mathcal{P}^{(O)}$ is the intersection of other three regular polygons $\mathcal{P}_1^{(O)}$, $\mathcal{P}_2^{(O)}$, $\mathcal{P}_3^{(O)}$, sharing the same center but with different apothems;
2. $\mathcal{P}^{(I)}$ consists in the union of two regular and polygons $\mathcal{P}_1^{(I)}$, $\mathcal{P}_2^{(I)}$ whose intersection is not empty;
3. $\mathcal{P}^{(I)} \subset \mathcal{P}^{(O)}$.

In general these operations are not easy to perform, since it is not straighforward to compute the intersection, the union and the subtraction of polygons, but the new release of Matlab R2017b, allows the solution of this problem via the built-in routines `intersect`, `union`, `subtract` applicable to polyshape objects. This allows to analyze complicated systems of lenses, even more general than ours and typical of optical design.

In the test, we consider regular polygons $\Omega^{(M_1)}$, $\Omega^{(M_2)}$

- $\mathcal{P}_1^{(O)}$ with center $C_1^{(O)} = (0, 0)$ and apothem $r_1^{(O)} = 1$,
- $\mathcal{P}_2^{(O)}$ with center $C_2^{(O)} = (0, 0)$ and apothem $r_2^{(O)} = 0.6120$,
- $\mathcal{P}_3^{(O)}$ with center $C_3^{(O)} = (0, -0.1184)$ and apothem $r_3^{(O)} = 0.5663$,
- $\mathcal{P}_1^{(I)}$ with center $C_1^{(I)} = (0, -0.1184)$ and apothem $r_1^{(I)} = 1.0761$,
- $\mathcal{P}_2^{(I)}$ with center $C_2^{(I)} = (0, -0.3761)$ and apothem $r_1^{(I)} = 1.2810$,

each one with vertices

$$C_k^{(s)} + r_k^{(s)}(\cos(t_j), \sin(t_j)),\ t_j = \frac{2j\pi}{M_i},\ j = 1, \ldots, M_i$$

being $s = O$ and $k = 1, 2, 3$ or $s = I$ and $k = 1, 2$ and M_i depending on $\Omega^{(M_i)}$ (i.e. 100 and 500). The polygons $\Omega^{(M_1)}$, $\Omega^{(M_2)}$, are partitioned respectively in 202 and 998 triangles.

These computations are performed to approximate integrals of certain algebraic polynomials necessary in the determination of lens aberration, and in general the degrees of exactness of interest are $\delta \leq 25$. We report that we experimented moments matching deterioration with the compression for higher degrees with Matlab NNLS code `lsqnonneg`, while these problems where not found with the open-source routine present in the package NNLSlab in [31].

TABLE 6.1

Comparison of the moments of cubature rules obtained via triangulation (with/without compression) as well as their cardinalities N_T, N_{TC}, on two polygons $\Omega^{(M_1)}$, $\Omega^{(M_2)}$, with ADE $= \delta$, $M_1 = 100$, $M_2 = 500$.

	$\Omega^{(M_1)}$				$\Omega^{(M_2)}$			
δ	$E_{T,TC}$	N_T	N_{TC}	Ratio$_{T,TC}$	$E_{T,TC}$	N_T	N_{TC}	Ratio$_{T,TC}$
5	$1e-15$	1414	21	67.3	$2e-15$	6986	21	332.7
10	$3e-15$	4848	66	73.5	$1e-14$	23952	66	362.9
15	$5e-15$	9292	136	68.3	$3e-15$	45908	136	337.6
20	$5e-15$	15756	231	68.2	$9e-15$	77844	231	337.0
25	$9e-15$	23634	351	67.3	$9e-15$	116766	351	332.7

All the numerical Matlab software is available at [31].

7. Acknowledgements. The work of A. Sommariva and M. Vianello is partially supported by the BIRD163015, PRAT-CPDA143275, DOR funds of the University of Padova, and by the GNCS-INdAM. Their research has been accomplished within the RITA *Research ITalian network on Approximation*.

REFERENCES

[1] F. BASAGLIA, A new cubature method over polygons (Italian), Laurea Thesis in Mathematics, University of Padova, 2009 (advisors A. Sommariva and M. Vianello).

[2] B. BAUMAN AND H. XIAO, Gaussian quadrature for optical design with noncircular pupils and fields, and broad wavelength range, Proc. SPIE, 7652(2) (2010), pp.1–12.

[3] M. BERN, J.R. SHEWCHUK, AND N. AMENTA, Triangulations and mesh generation, Handbook of Discrete and Computational Geometry, Third Edition, CRC Press.

[4] J. BERNTSEN, T.O. ESPELID, Algorithm 706: DCUTRI: an algorithm for adaptive cubature over a collection of triangles, ACM Transactions on Mathematical Software, 18 (1992), pp.329–342.

[5] P.A. BUSINGER AND G.H. GOLUB, Linear least-squares solutions by Householder transformations, Numer. Math. 7 (1965), pp.269–276.

[6] M.G. BLYTH AND C. POZRIKIDIS, A Lobatto integration grid over the triangle, IMA J. Appl. Math (2005), pp. 1–17.

[7] C. CARATHEODORY, Über den Variabilittsbereich der Koeffizienten von Potenzreihen, die gegebene Werte nicht annehmen, Math. Ann. 64 (1907), pp.95–115.

[8] B. CHAZELLE, Triangulating a Simple Polygon in Linear Time, Discrete and Computational Geometry, 6 (1991), pp.485–524.

[9] R. COOLS, Constructing cubature formulae: the science behind the art, Volume 6 January 1997, pp.1–54.

[10] G. DA FIES, A. SOMMARIVA AND M.. VIANELLO, Algebraic cubature by linear blending of elliptical arcs, Appl.Num.Math. 74 (2013), pp.49–61.

[11] E. DE DONCKER AND I. ROBINSON, An Algorithm for Automatic Integration over a Triangle Using Nonlinear Extrapolation, ACM Trans. Math. Soft., 10 (1984), pp.1–16.

[12] D.A. DUNAVANT, High degree efficient symmetrical gaussian quadrature rules for triangles, Int. J. for Num. Meth. Eng., 21 (1985), pp.1129–1148.

[13] M. FESTA AND A. SOMMARIVA, Computing almost minimal formulas on the square, J. Comput. Appl. Math. 236 (2012), pp.4296–4302.

[14] G.W. FORBES, Optical system assessment for design: numerical ray tracing in the Gaussian pupil, J. Opt. Soc. Am. A 5 (1988), pp.1943–1956.

[15] M. GENTILE, A. SOMMARIVA AND M. VIANELLO, Polynomial interpolation and cubature over polygons, J. Comput. Appl. Math. 235 (2011), pp.5232–5239.

[16] M. GENTILE, A. SOMMARIVA AND M. VIANELLO, POLYGINT: a Matlab code for interpolation and cubature at Discrete Extremal Sets of polygons (available online at: `www.math.unipd.it/~marcov/CAAsoft.html`).

[17] A. GENZ, An adaptive numerical integration algorithm for simplices, Lecture Notes in Computer Science, 507, Springer-Verlag, (1991), pp.279–285.

[18] D.K. Kahaner, O.W. Rechard, TWODQD an adaptive routine for two-dimensional integration, J. Comput. Appl. Math. 17 (1987), pp.215–234.
[19] P. Keast, Software for Integration Over Triangles and General Simplices, NATO ASI Series (Series C: Mathematical and Physical Sciences), vol 357 (1992), pp..283-294.
[20] D.P. Laurie, CUBTRI: Automatic Integration over a Triangle, ACM Trans. Math. Soft., 8 (1982), pp.210–218.
[21] C.L. Lawson and R.J. Hanson, Solving least squares problems. Revised reprint of the 1974 original, Classics in Applied Mathematics 15, SIAM, Philadelphia, 1995.
[22] J.N. Lyness and R. Cools, A Survey of Numerical Cubature over Triangles, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill, 1994.
[23] J.N. Lyness, D. Jespersen, Moderate Degree Symmetric Quadrature Rules for the Triangle, J. Inst. Maths Applies, 15 (1975), pp.19-32.
[24] Mathworks, Polyshape documentation, (available in Matlab R2017b and online at: `https://it.mathworks.com/help/matlab/ref/polyshape.html`).
[25] G.H. Meisters, Polygons have ears, American Mathematical Monthly, (1975), pp.648–651.
[26] S-A. Papanicolopulos, Computation of moderate-degree fully symmetric cubature rules on the triangle using symmetric polynomials and algebraic solving, Comput. Math. Appl. 69 (2015), pp.650–666.
[27] S-A. Papanicolopulos, New fully symmetric and rotationally symmetryc cubature rules on the triangle using minimal orthonormal bases, J. Comput. Appl. Math. 294 (2016), pp.39–48.
[28] F. Piazzon, A. Sommariva and M. Vianello, Caratheodory-Tchakaloff Subsampling, Dolomites Res. Notes Approx. DRNA 10 (2017), pp.5–14.
[29] L. Shampine, Vectorized adaptive quadrature in Matlab, J. Comput. Appl. Math., 211 (2008), pp.131–140.
[30] M. Slawski, Non-negative least squares: comparison of algorithms, `https://sites.google.com/site/slawskimartin/code`.
[31] A. Sommariva, `http://www.math.unipd.it/∼alvise/software.html`.
[32] A. Sommariva, `http://www.math.unipd.it/∼alvise/sets.html`.
[33] A. Sommariva and M. Vianello, Product Gauss cubature over polygons based on Green's integration formula, BIT Numerical Mathematics 47 (2007), pp.441–453.
[34] A. Sommariva and M. Vianello, Compression of multivariate discrete measures and applications, Numer. Funct. Anal. Optim. 36 (2015), pp.1198–1223.
[35] A. Sommariva and M. Vianello, Nearly optimal nested sensors location for polynomial regression on complex geometries, to appear on Sampling Theory in Signal and Image Processing.
[36] S. Suri, Polygon Triangulation, (available online at: `https://www.cs.ucsb.edu/∼suri/cs235/Triangulation.pdf`).
[37] M.A. Taylor, B.A. Wingate, L.P. Bos, A cardinal function algorithm for computing multivariate quadrature points, SIAM Journal on Numerical Analysis, volume 45, no. 1 (2007), pp.193–205.
[38] V. Tchakaloff, Formules de cubatures mécaniques à coefficients non négatifs, (French) Bull. Sci. Math. 81 (1957), pp.123–134.
[39] M. Tchernychova, *Caratheodory* cubature measures, Ph.D. dissertation in Mathematics (supervisor: T. Lyons), University of Oxford, 2015.
[40] R.T. Trîmbitas, Adaptive Cubatures on Triangle, Results. Math.,, 53 (2009), pp.453?462,
[41] H. Xiao and Z. Gimbutas, A numerical algorithm for the construction of efficient quadrature rules in two dimensions, Comput. Math. Appl. 59 (2010), pp.663–676.
[42] H. Xiao, V. Rokhlin and N. Yarvin, Prolate SpheroidalWave Functions, Quadrature and Interpolation, Inverse Problems 17 (2001), pp.805–838.
[43] S. Wandzura and H. Xiao, Symmetric Quadrature Rules on a Triangle, Comput. Math. Appl. 45 (2003), pp.1829–1840.