Compressed cubature over polygons with applications to optical design

B. Bauman[1], A. Sommariva[2] and M. Vianello[2]

December 5, 2019

Abstract

In this paper we propose an algorithm to determine cubature rules of algebraic degree of exactness δ on general polygons $\mathcal{P}$, by means of Matlab `polyshape` objects and near minimal rules on triangles, obtaining by Caratheodory-Tchakaloff subsampling a PI (Positive Interior) final formula with cardinality at most $N_\delta = (\delta+1)(\delta+2)/2$. We test our algorithm on polygons with different shape, and we also discuss an application to the computation of the RMSWE (Root Mean Square Wavefront Error) on obscured and vignetted pupils, in the framework of optical design by numerical ray tracing for the LSST (Large Synoptic Survey Telescope).

2010 AMS subject classification: 65D32, 78M25.

Keywords: numerical cubature, polygonal domains, minimal triangulation, minimal formulae, Caratheodory-Tchakaloff subsampling, optical design, numerical ray tracing, obscured and vignetted pupils.

1 Introduction

In this paper we propose an algorithm that computes low-cardinality cubature rules with Algebraic Degree of Exactness $ADE = \delta$ on a polygon $\mathcal{P}$, that can be defined by union, intersection and difference of other polygons, and can be in general non-simple, multiply-connected or even disconnected. These rules are of PI-type, where P stands for *Positive* weights and I for *Interior* nodes (i.e., all the cubature nodes are in the domain).

As a general strategy, we first construct and triangulate the polygon by the Matlab `polyshape` routines, obtaining a number of triangles that is minimal or almost minimal, and then we take a near minimal rule of PI-type with $ADE = \delta$ on each triangle, thus determining a PI cubature rule on the whole $\mathcal{P}$. Similar techniques are not new, but the present approach has the advantage of treating very general polygons (to our knowledge this is the first time that Matlab `polyshape` is used in a cubature package), as well as of using basic cubature formulae whose cardinality is usually lower than that offered by other general purpose codes.

[1] Lawrence Livermore National Laboratory, Livermore, USA

[2] Department of Mathematics, University of Padova, Italy
corresponding author: marcov@math.unipd.it

A further relevant improvement is given by compression of the cubature formula into at most $N_\delta = (\delta + 1)(\delta + 2)/2$ cubature nodes and positive weights, by Caratheodory-Tchakaloff subsampling of discrete measures. This step consists essentially in computing a nonnegative sparse solution to the corresponding underdetermined moment system for the weights, which exists in view of a discrete version of Tchakaloff's Theorem, by standard mathematical programming methods; cf. [18, 24] and the references therein.

We discuss a number of numerical tests, on polygons with different shape: convex, concave, non-simple, multiply-connected. In particular, we apply the method to the computation of the RMSWE (Root Mean Square Wavefront Error) on a circular telescope pupil obscured and vignetted by several co-axial disks (approximated by regular polygons with hundreds of sides), a problem arising in optical design by numerical ray tracing for the LSST (Large Synoptic Survey Telescope, cf. [1, 16]).

2 About the subdivision

One of the key points to determine a good cubature rule over a polygon $\mathcal{P}$, consists in partitioning $\mathcal{P}$ into simpler regions $\Omega_1, \dots, \Omega_\nu$ and then applying a known low-cardinality formula on each of them.

In [23], the authors considered the case of *simple* polygons, i.e. with non self-intersecting boundary path, 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, 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 of *triangulators*, a PI formula can be obtained via Gauss-Green theorem together with univariate Gaussian quadrature. If the baseline property does not hold, a cubature rule can still be computed by another reference line, but without the warranty of being of PI-type (some nodes can fall outside the domain and some weights can be negative).

In order to enlarge the class of polygons for which a rule of PI-type can be constructed, one can resort to minimal triangulation or quadrangulation algorithms, cf. e.g. [11]. It is known indeed that any simple and simply connected polygon with n vertices can be partioned into $\nu = n - 2$ triangles, or into $(n-2)/2 \leq \nu \leq n-2$ quadrangles (some possibly degenerating into triangles) where ν is often close to the lower bound. For example, any convex polygon is trivially partitioned into $\nu = (n-2)$ triangles by fixing a vertex and connecting it to the non adjacent ones, or into $\nu = (n-2)/2$ quadrangles for n even and $\nu = (n-3)/2$ quadrangles plus one triangle for n odd, simply by taking quadruples of consecutive vertices. In general, while on simple and simply connected polygons minimal (but very complicated) triangulation algorithms with $\mathcal{O}(n)$ complexity are known, the most popular ones have typically a $\mathcal{O}(n \log n)$ complexity, cf. e.g. [4, 26]. Then by rules that are near minimal and of PI-type on the simplex or alternatively of product Gaussian type, one can easily achieve rules on each triangular or quadrangular subdomain of the partition, and eventually a PI rule on the whole $\mathcal{P}$.

Working with polygons has become much simpler in Matlab, after the introduction of 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 any polygon $\mathcal{P}$, that can be of very general nature, even non-simple, or multiply-connected or disconnected (we notice however that Matlab does not provide any reference about the triangulation algorithms). We tested the quality of the triangulation $\{\Omega_i\}$ on very different polygonal domains $\mathcal{P}$ with n sides, achieving $\mathcal{P} = \cup_{i=1}^{\nu} \Omega_i$ with $\nu \approx n$ and very often $\nu = n - 2$ (minimal triangulation) in the case of simple and 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; the tests have been performed in Matlab R2017b on a 2,7 GHz Intel Core i5 with 16 GB 1867 MHz DDR3 memory. As regions, given n equally spaced angles in $[0, 2\pi]$ say $t_k = \frac{2\pi k}{n}$, with $k = 1, \ldots, n$, we have taken (see Figure 1)

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

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

2. a polygonal cardioid $\mathcal{P}^{(2)}$ whose n vertices are

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

3. a polygonal Bernoulli lemniscate $\mathcal{P}^{(3)}$ whose n vertices are

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

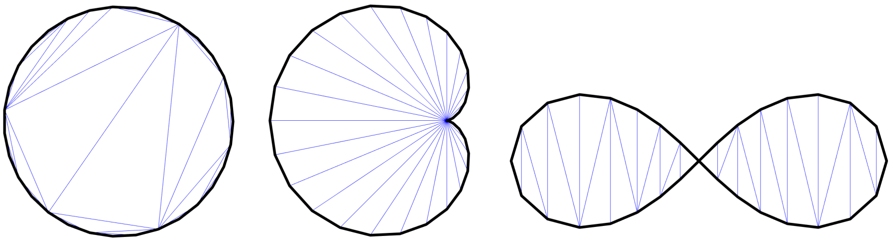

Figure 1: From left to the right, the three polygons $\mathcal{P}^{(1)}$, $\mathcal{P}^{(2}$, $\mathcal{P}^{(3)}$ for $n = 32$, and their triangulation.

Observe that the polygonal Bernoulli Lemniscate is non-simple, due to the self intersection in the origin, but can be still correctly treated up to $n = 1000$. 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 the three polygons $\mathcal{P}^{(1)}$, $\mathcal{P}^{(2)}$ and $\mathcal{P}^{(3)}$ are listed in Table 1, respectively as t_P and t_T, an asterisk meaning that the triangulation process presents warnings. The results show 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.

Table 1: Average triangulation cputimes on several polygonal domains; an asterisk means that the triangulation process presents warnings.

	$\mathcal{P}^{(1)}$		$\mathcal{P}^{(2)}$		$\mathcal{P}^{(3)}$	
vertices	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*

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 PI cubature rule on $\mathcal{P}$, by the additivity of the integral operator, it is sufficient to define a PI rule on each subdomain Ω_i. In the present paper we shall focus on triangulations.

Now, since any triangle is bijectively mapped into any other by an affine transformation, knowing a cubature rule on a reference triangle $\mathcal{T}^*$, after the conversion of the nodes in barycentric coordinates, it is straightforward to achieve one on each other possible triangle $\mathcal{T}$ by varying the weights proportionally to their area. In other words, let $\mathcal{T}^*$ be the unit simplex with vertices $(0,0)$, $(0,1)$, $(1,0)$, $\{\phi_k\}_{k=1,\dots,N_\delta}$ be a basis of the space of bivariate polynomials $\mathbb{P}_\delta$ of total degree not exceeding δ and cardinality $N_\delta = (\delta+1)(\delta+2)/2$, and assume that

$$\int_{\mathcal{T}^*} \phi_k(x)\,dx = \sum_{i=1}^{m} w_i^* \phi_k(\xi_i^*) \ , \ \xi_i^* \in \mathcal{T}^* \ , w_i^* > 0 \ , \ \ k = 1, \dots, N_\delta \ , \tag{1}$$

i.e. that the PI cubature rule $\{(\xi_i^*, w_i^*)\}$ has $ADE = \delta$.

Now, let $\xi_i^* = (x_i^*, y_i^*)$ and denote by $(x_i^*, y_i^*, 1 - x_i^* - y_i^*)$ 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

$$\xi_i = x_i^* V_1 + y_i^* V_2 + (1 - x_i^* - y_i^*) V_3, \quad i = 1, \dots, m$$

and weights

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

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 cubature rules with a certain ADE only on the reference triangle $\mathcal{T}^*$. To his purpose, the more appealing ones are the so called *minimal rules*, that have the lowest number of nodes between PI rules with $ADE = \delta$. Differently from the univariate case, where they are the so-called *Gaussian rules*, only few multivariate minimal rules are known; see e.g. [5].

In spite of this, the bibliography about PI rules with low cardinality for a given ADE, say *near minimal* rules, is rather wide. A common technique for their determination is the following. If the basis moments $\gamma_k = \int_{\mathcal{T}^*} \phi_k(x)\,dx$, $k = 1, \ldots, N_\delta$ are known, one computes via numerical optimization the solutions (ξ_i, w_i), with $i = 1, \ldots, m$ of the nonlinear problem (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., [27]). 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 strategies provided many near minimal pointsets, even for rather large δ (see e.g. [14, 17, 30, 32] with the references therein).

In Table 2 we have listed to the authors' knowledge the best of these near minimal rules, that are used to implement a PI formula with $ADE = \delta$ on any triangle and eventually on any triangulated polygon $\mathcal{P}$. Observe that one can use any triangulation to determine a PI rule on $\mathcal{P}$, but a minimal triangulation has the advantage of keeping as low as possible the cardinality, since in any triangle a fixed number of points is used.

All the minimal rules are stored as a Matlab file at [22], where we have corrected the formulae whenever necessary in order to match (1) close to machine precision. For higher degrees one can use the well-known Stroud conical rules [13], whose cardinality is $(\delta + 1)^2/4$ for odd δ.

Table 2: Cardinality N_δ^* of (near) 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

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 one with the same degree of exactness, positive weights, but with cardinality equal at most to the dimension N_δ of the polynomial space $\mathbb{P}_\delta$.

In some recent papers, the authors have applied a mathematical method named CATCH (acronym of "Caratheodory-Tchakaloff" subsampling), for the 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 theoretical tool is the following discrete version of the well-known Tchakaloff theorem [28], that can be proved by Caratheodory theorem on finite-dimensional conic/convex combinations [3]; cf. [18] and the references therein.

Theorem 1 *Let μ be a (multivariate) measure whose support is a $\mathbb{P}_\delta$-determining finite set $X = \{\xi_i\} \subset \mathbb{R}^k$ (i.e., a polynomial in $\mathbb{P}_\delta$ vanishing there vanishes everywhere in $\mathbb{R}^k$), with corresponding positive weights (masses) $\lambda = \{\lambda_i\}$, $i = 1, \ldots, M$, $M = card(X) > N_\delta = dim(\mathbb{P}_\delta) = \binom{\delta+k}{k}$. Then, there exist a cubature formula for the discrete measure μ, with nodes $T_\delta = \{t_j\} \subset X$ and positive weights $w = \{w_j\}$, $1 \le j \le m$, with $m \le N_\delta$, such that*

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

An interpretation of this theorem in the framework of cubature is:

- if we have at hand a PI rule, say (X, λ), with $ADE = \delta$ and cardinality $M > N_\delta$, then we can "compress" it into a PI rule, say (T_δ, w), with cardinality not exceeding N_δ.

In order to implement the result, given any polynomial basis $\{\phi_k\}$ of $\mathbb{P}_\delta$, define the Vandermonde-like matrix

$$V = V_n(X) = \{\phi_j(\xi_i)\} \ , \ \ 1 \le i \le M \ , \ 1 \le j \le N_\delta \ , \tag{2}$$

let $\gamma = V^T\lambda$ the vector of moments of the polynomial basis $\{\phi_j\}$ with respect to the original discrete measure and consider the underdetermined moment system $V^T u = \gamma$. Theorem 1 then asserts that there exists a sparse nonnegative solution u^* to such a system, whose nonvanishing components (i.e., the weights $\{w_j\}$) are at most N_δ and determine the corresponding reduced sampling points $T_\delta = \{t_j\} \subseteq X$, that we may term Caratheodory-Tchakaloff (CATCH) points of X.

To our knowledge, essentially two approaches have been developed to get these compressed rules, i.e. via Linear Programming (LP) via and Quadratic Programming (QP); cf. [18, 24, 29] and the references therein.

Concerning the LP approach, it consists in solving (via the simplex-method)

$$\left\{ \begin{array}{l} \min c^T u \\ V^T u = \gamma \ , \ \ u \ge 0 \ , \end{array} \right. \tag{3}$$

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 [18, 29].

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

$$\|V^T u^* - \gamma\|_2 = \min\{\|V^T u - \gamma\|_2 \ , \ u \geq 0\} \ , \tag{4}$$

in which u^* can be obtained by the well-known *Lawson-Hanson* active set optimization method [12], which determines a sparse solution to (4). Its application gives a residual $\varepsilon = \|V^T u^* - \gamma\|_2$ that is extremely small, say $< 10^{-14}$ for $\delta \leq 30$.

Our numerical experience in bivariate instances with presently available Matlab software has shown that NNLS usually performs better than LP in computing the CATCH weights, at least for moderate degrees δ, cf. [18]. Consequently all our codes are based on the application of Lawson-Hanson method to compute the Caratheodory-Tchakaloff compressed rules.

We point out that there are several versions of NNLS codes available in Matlab. One is the built-in function `lsqnonneg`, based on (a variant of) the Lawson-Hanson algorithm while an open-source version is present in the package NNLSlab in [21]. Other implementations are often obtained by MEX files and will not be used here, since we wish to provide codes that can be immediately used by standard Matlab users.

It should also be recalled that an algorithm termed *Recursive Halving Forest*, based on a hierarchical SVD, has been proposed in [29] to compute the compressed cubature nodes and weights. 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 Examples and applications

In order to make some numerical tests, we first compare the results that we can obtain via this new algorithm with those in [23], and next we show the effect of Caratheodory-Tchakaloff subsampling in the compression of PI polygonal cubature rules on polygons with different shape. Moreover, we propose a method to obtain embedded rules on general polygonal domains.

Finally, we discuss an application in optics, namely the computation of the RMSWE (Root Mean Square Wavefront Error) on a telescope pupil obscured and vignetted by several co-axial disks (approximated by regular polygons with a very large number of sides). This application is relevant to optical design by numerical ray tracing for the LSST (Large Synoptic Survey Telescope).

5.1 Numerical tests

As in Section 1, all the tests have been performed in Matlab R2017b on a 2,7 GHz Intel Core i5 with 16 GB 1867 MHz DDR3 memory.

In [23], the method was tested on a convex and on certain concave polygon, say $\mathcal{P}_{conv}$ and $\mathcal{P}_{conc}$, achieving PI rules (see Figure 2 for the description of the polygons, baselines and pointset distribution for algebraic degree of exactness $\delta = 10$).

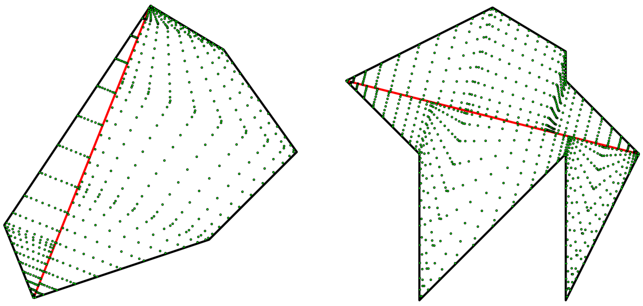

Figure 2: The pointsets for $ADE = \delta = 10$ obtained by [23] on $\mathcal{P}_{conv}$ and $\mathcal{P}_{conc}$ (in red the baseline).

In Table 3, we compare the cubature cardinality $M_\delta^{(old)}$ of the approach in [23], with those of the present one before and after compression, say $M_\delta^{(new)}$ and m_δ respectively. 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 polynomial space $\mathbb{P}_\delta$, and that the compression ratio $M_\delta^{(new)}/m_\delta$ depends on the number of sides of the polygon.

Table 3: Cardinalities $M_\delta^{(old)}$, $M_\delta^{(new)}$, m_δ, applying the algorithm proposed in [23] and the one presented here before and after compression, on the two polygons of Figure 2 with $ADE = \delta = 5, 10, \ldots, 40$.

	$\mathcal{P}_{conv}$			$\mathcal{P}_{conc}$		
δ	$M_\delta^{(old)}$	$M_\delta^{(new)}$	m_δ	$M_\delta^{(old)}$	$M_\delta^{(new)}$	m_δ
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

Concerning the computational cost, determination of the two basic rules rules is not an issue since it takes only a small fraction of the overall cputime (less than 1% for δ in the order of tens). Compression using the Matlab built-in routine `lsqnonneg` or the open-source NNLSlab is in general not too time consuming for low degrees, while it becomes relevant (order of 10^1 seconds, more than 99% of the overall cputime) for $\delta > 30$.

In order to control ill-conditioning (which arises already at small exactness degrees with the standard monomial basis), we used as proposed in [24] the total-degree product Chebyshev basis $\{T_p(\alpha_1(x))T_q(\alpha_2(y))\}$, $0 \leq p+q \leq \delta$, $(x, y) \in [a_1, b_1] \times [a_2, b_2]$ (the smallest cartesian rectangle containing the polygonal domain), where $T_h(\cdot) = \cos(h \arccos(\cdot))$ is the h-degree Chebyshev polynomial and $\alpha_i(s) = (2s - b_i - a_i)/(b_i - a_i)$, $s \in [a_i, b_i]$, $i = 1, 2$. For the compression we used in particular the software available from the package NNLSlab in [21].

The relative gaps between the bivariate Chebyshev moments computed by the Gauss-Green-based rule of [23], the present triangulation-based rule and the compressed rule turn out to be not far from machine precision, say around 10^{-14}-10^{-15}, in all the tests.

To show the advantages of this new approach via `polyshape` triangulation, we consider as examples two polygons $\mathcal{P}_1$ and $\mathcal{P}_2$, that we can think as discretization of a quatrefoil, which could not be treated by the previous algorithms since the domains are non-simple and have not the axis property.

More precisely, the polygon vertices are $(\cos(t_k)\cdot\sin(2t_k), \sin(t_k)\cdot\sin(2t_k)))$, where $t_k = \frac{2k\pi}{M_i}$, with $k = 1,\ldots,M_i$, $i = 1,1$. In particular, we set $M_1 = 129$, and $M_2 = 513$. The polygon $\mathcal{P}_1$ can be partioned in $N_{tri} = 120$ triangles, while $\mathcal{P}_2$ in $N_{tri} = 504$ triangles.

Table 4: Cardinalities $M_\delta^{(new)}$, $\tilde{M}_\delta^{(new)}$ of cubature rules obtained via triangulation (with/without compression) on two non-simple polygons $\mathcal{P}_1$ and $\mathcal{P}_2$ approximating a quatrefoil, with $ADE = \delta = 5, 10, \ldots, 35$.

	$\mathcal{P}_1$			$\mathcal{P}_2$		
δ	$M_\delta^{(new)}$	$\tilde{M}_\delta^{(new)}$	Ratio$_\delta$	$M_\delta^{(new)}$	$\tilde{M}_\delta^{(new)}$	Ratio$_\delta$
5	840	21	40.0	3528	21	168.0
10	2880	66	43.6	12096	66	183.3
15	5520	136	40.6	23184	136	170.5
20	9360	231	40.5	39312	231	170.2
25	14040	351	40.0	58968	351	168.0
30	20520	496	41.4	86184	496	173.8
35	27360	666	41.1	114912	666	172.5

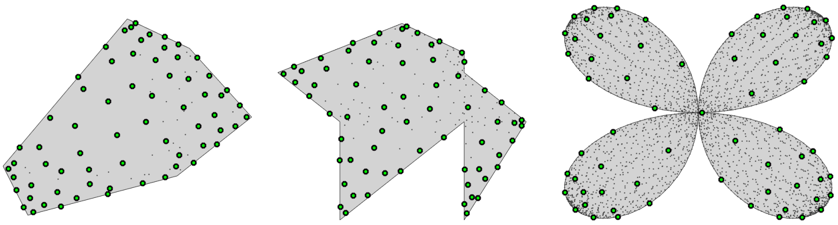

Figure 3: Compressed set at degree $\delta = 10$, for three polygons with different shape: convex, concave, non-simple (quatrefoil).

By a careful look at Table 4, we observe that the compression ratio 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 (being near minimal), thus $M_\delta^{(new)} \approx N_{tri}\delta^2/6$ and since the compressed set has cardinality $\tilde{M}_\delta^{(new)} \approx \delta^2/2$, necessarily

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

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

$$Q^{(L)}(f) = \sum_{j=1}^{M_L} w_j^{(L)} f(x_j^{(L)}),\ Q^{(H)}(f) = \sum_{s=1}^{M_H} w_s^{(H)} f(x_s^{(H)})$$

with different degree of exactness, say $\delta_{Q^{(L)}} < \delta_{Q^{(H)}}$, where $\{x_j^{(L)}\} \subset \{x_s^{(H)}\}$ (see also [25]). A typical application is that they allow stopping criteria, based on rules with different degrees of exactness δ_1, δ_2, with $\delta_1 < \delta_2$, with the property that the nodes of the rule with $ADE = \delta_1$ are among 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; cf., e.g., [6] on the concept of embedded cubature rule.

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

$$\mathcal{P} \supset X \supset T_{\delta_k} \supset T_{\delta_{k-1}} \supset \cdots \supset T_{\delta_2} \supset T_{\delta_1} \tag{5}$$

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

$$\text{compute } u_{j-1}^* \ :\ \|A_j u_{j-1}^* - b_j\|_2 = \min \|A_j u_{j-1} - b_j\|_2 \ ,\ \ u_{j-1} \geq 0 \ , \tag{6}$$

for $j = k+1, k, \ldots, 2$, where $A_j = (V_{\delta_{j-1}}(T_{\delta_j}))^t$ (cf. (2)), $b_j = A_j u_j^*$, and we set $T_{\delta_{k+1}} = X$, $u_{k+1}^* = \lambda$, (X, λ) being the starting cubature formula.

5.2 An application to optical design

As application of these new cubature formulae, we consider a problem arising in optical design. The use of low-cardinality cubature in optical design began with a classical paper by Forbes [9], who suggested to adopt product Gaussian quadrature for efficient numerical ray tracing on circular or elliptical apertures (pupils), to compute the Root Mean Square (RMS) Spot Size. Later, in [1] the authors studied cubature methods based on prolate spheroidal wave functions and Gaussian quadrature [31], to treat situations where the pupil is obscured and vignetted (a feature that occurs, for example, in optical astronomy; see Figure 4). We recall that such Gaussian quadrature approaches are currently implemented within one of the most popular optical design software suites, the Zemax OpticsStudio package [34], to compute for example one of the relevant optical parameters, the Root Mean Square Wavefront Error (RMSWE, cf. e.g. [19]), that is

$$\mathrm{RMSWE}_\Omega = \left(\int_\Omega W^2(x) \, \frac{dx}{\mathcal{A}} - \left(\int_\Omega W(x) \, \frac{dx}{\mathcal{A}} \right)^2 \right)^{1/2} \ ,\ \ \mathcal{A} = area(\Omega) \ , \tag{7}$$

where Ω is the pupil integration region and W the optical wavefront, usually approximated by a low-degree truncated Zernike expansion.

In [1] the authors considered a 3-disks model for a LSST-like (Large Synoptic Survey Telescope, [16]) aperture, where the main circular pupil 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), by two co-axial disks.

The same model was then treated in [7] by subperiodic trigonometric Gaussian quadrature, via domain splitting into circular (even asymmetric) sectors and a truncated annulus.

The configurations become more complicated with the 5-disks model coped here, that could not be treated by efficient numerical cubature within the presently available optical design packages. In this model obscuration and vignetting are made by four co-axial disks, see Figure 4. The relevant region is obtained by intersection with two larger external disks and subtraction of two internal disks. There are several configurations, depending on the mutual position of the co-axial disks centers and on the disk radii. All the disks are approximated by regular polygons with a very large number of sides, so that the non obscured region is polygonal and easily obtainable by Matlab `polyshape` and the built-in routines `intersect`, `union` and `subtract`.

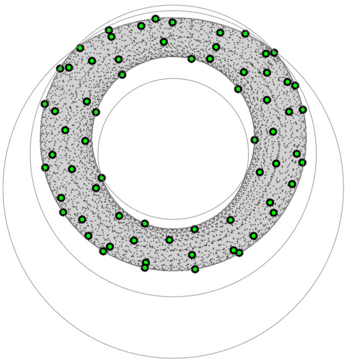

Figure 4: Cubature nodes for an obscured and vignetted telescope pupil at exactness degree $\delta = 10$; original formula (dots) and compressed formula (small circles).

In the numerical example the main telescope pupil is the unit disk, the two external clipping disks have centers $(0, -0.1184)$, $(0, -0.3761)$ and radii 1.0761, 1.2810, whereas the two internal obscuring disks have centers $(0, 0)$, $(0, -0.1184)$ and radii 0.6210, 0.5663, respectively. The five circles have been approximated by regular polygons with doubled sides from 100 up to 1600, obtaing an approximate polygonal integration region, say $\mathcal{P} \approx \Omega$ (in grey in Figure 4 corresponding to 800 sides per circle).

The integration test has been performed taking as waveform $W(x)$ a linear combination with uniform random coefficients of the first nine Zernike polynomials $Z_0, \dots, Z_8$, as listed in [33], which allow to take into account the main classical optical aberrations; thus, the resulting wavefront is a polynomial of degree 4, and the integrand in (7) is a polynomial of degree $\delta = 8$.

In Table 5 we show the average relative error over 1000 trials of the random array of nine Zernike coefficients

$$\mathcal{E}_{rel} = \frac{|\mathrm{RMSWE}_{\Omega} - \mathrm{RMSWE}_{\mathcal{P}}|}{\mathrm{RMSWE}_{\Omega}} \,,$$

where RMSWE$_{\mathcal{P}}$ has been computed by the compressed polygonal PI formula discussed in the present paper for $ADE = 8$ (45 points), whereas the reference value RMSWE$_{\Omega}$ has been computed at machine precision via algebraic cubature with $ADE = 8$ on set difference of disks (lunes) followed by subtraction of such integrals on lunes from the integral on the underlying circular pupil, using the formulae developed in [8] (we stress that such an approach, considered as a unique cubature formula, would present some non-interior points and some negative weights).

Table 5: Cubature construction cputime and average relative error over 1000 trials in computing the RMSWE for a random linear combination of the first nine Zernike polynomials on the obscured and vignetted polygonal pupil of Figure 4 (the triangulation-based starting cubature formula is compressed into 45 nodes and positive weights).

sides per circle	100	200	400	800	1600
pupil sides	204	403	800	1597	3188
card. triang.-based	3232	6416	12768	25520	50976
cubat. constr. time	0.06s	0.09s	0.16s	0.27s	1.00s
average $\mathcal{E}_{rel}$	2.9e-03	7.4e-04	1.8e-04	4.5e-05	1.1e-05

We see that error is of the order of $1/L^2$, where L is the number of sides of the regular polygons approximating the disks, which is not surprising since this is the expected error order on the pupil area. Already for L in the hundreds this guarantees a reasonably accurate approximation of the RMSWE (error of about 0.01%), with a low computational cost. Finally, it is worth observing that the present approach, based on polygonal boundary approximation, could be easily applied to much more complicated pupil masking configurations of interest in astronomical optics.

6 Acknowledgements

The work of A. Sommariva and M. Vianello is partially supported by the projects BIRD181249, BIRD192932 and the 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] B. Bauman and H. Xiao, Gaussian quadrature for optical design with noncircular pupils and fields, and broad wavelength range, Proc. SPIE, 7652(2) (2010), 1–12.

[2] M.G. Blyth and C. Pozrikidis, A Lobatto integration grid over the triangle, IMA J. Appl. Math (2005), 1–17.

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

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

[5] R. Cools, Constructing cubature formulae: the science behind the art, Acta Numer. 6 (1997), 1–54.

[6] R. Cools, Monomial cubature rules since “Stroud”: a compilation - part 2, J. Comput. Appl. Math. 112 (1999), 21–27.

[7] G. Da Fies, A. Sommariva and M. Vianello, Algebraic cubature by linear blending of elliptical arcs, Appl.Num.Math. 74 (2013), 49–61.

[8] G. Da Fies and M. Vianello, Product Gaussian quadrature on circular lunes, Numer. Math. Theory Methods Appl. 7 (2014), 251–264.

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

[10] P. Keast, Software for Integration Over Triangles and General Simplices, NATO ASI Series (Series C: Mathematical and Physical Sciences), vol 357 (1992), 283–294.

[11] J.M. Keil, Polygon Decomposition, Ch. 11 in: Handbook of Computational Geometry, J.-R. Sack and J. Urrutia Eds., Elsevier, 2000, 491–518.

[12] 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.

[13] J.N. Lyness and R. Cools, A survey of numerical cubature over triangles, in: Proc. Sympos. Appl. Math., 48, AMS, Providence, 1994, 127–150.

[14] J.N. Lyness, D. Jespersen, Moderate Degree Symmetric Quadrature Rules for the Triangle, J. Inst. Maths Applies 15 (1975), 19–32.

[15] Mathworks, Polyshape documentation, available from Matlab R2017b and online at:
`https://it.mathworks.com/help/matlab/ref/polyshape.html`.

[16] S.S. Olivier, L. Seppala and K. Gilmore, Optical design of the LSST camera, Proc. SPIE 7018 (2008).

[17] S.-A. Papanicolopulos, New fully symmetric and rotationally symmetryc cubature rules on the triangle using minimal orthonormal bases, J. Comput. Appl. Math. 294 (2016), 39–48.

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

[19] R.R. Shannon, Handbook of Optics - Chapter 35, Optical Specifications, McGraw-Hill, New York, 1995.

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

[21] A. Sommariva, Cubature codes, https://www.math.unipd.it/\sim$alvise/software.html.

[22] A. Sommariva, Cubature data sets, https://www.math.unipd.it/\sim$alvise/sets.html.

[23] A. Sommariva and M. Vianello, Product Gauss cubature over polygons based on Green's integration formula, BIT Numerical Mathematics 47 (2007), 441–453.

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

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

[26] S. Suri, Polygon Triangulation, https://www.cs.ucsb.edu/~suri/cs235/Triangulation.pdf.

[27] 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), 193–205.

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

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

[30] H. Xiao and Z. Gimbutas, A numerical algorithm for the construction of efficient quadrature rules in two dimensions, Comput. Math. Appl. 59 (2010), 663–676.

[31] H. Xiao, V. Rokhlin and N. Yarvin, Prolate Spheroidal Wave Functions, Quadrature and Interpolation, Inverse Problems 17 (2001), 805–838.

[32] S. Wandzura and H. Xiao, Symmetric Quadrature Rules on a Triangle, Comput. Math. Appl. 45 (2003), 1829–1840.

[33] J.C. Wyant, Zernike Polynomials for the Web, 2003 (Adaptive Optics - Max Planck Institute for Astronomy) www.mpia.de/AO/INSTRUMENTS/FPRAKT/ZernikePolynomialsForTheWeb.pdf.

[34] Zemax, OpticStudio package, www.zemax.com/products/opticstudio.