Numerical hyperinterpolation over nonstandard planar regions*

Alvise Sommariva and Marco Vianello[1]

August 3, 2016

Abstract

We discuss an algorithm (implemented in Matlab) that computes numerically total-degree bivariate orthogonal polynomials (OPs) given an algebraic cubature formula with positive weights, and constructs the orthogonal projection (hyperinterpolation) of a function sampled at the cubature nodes. The method is applicable to nonstandard regions where OPs are not known analytically, for example convex and concave polygons, or circular sections such as sectors, lenses and lunes.

2000 AMS subject classification: 41A63, 42C05, 65D10, 65D32

Keywords: multivariate orthogonal polynomials, positive cubature, hyperinterpolation, nonstandard planar regions.

1 Introduction

Orthogonal polynomials (OPs) are an important area of applied and computational mathematics. In the one-dimensional case, the subject has reached a substantial maturity, that from the numerical point of view is well represented, for example, by Gautschi's reference work on computation and approximation of OPs and related methods, such as Gaussian quadrature, together with the OPQ Matlab suite; cf. [14].

In the multivariate case, the subject has been knowing recently a fast progress, especially on the theoretical side, with an important reference in the seminal book by Dunkl and Xu [13], but is still far from maturity. Computation of multivariate OPs (and related orthogonal projections/expansions) over nonstandard regions is still a challenging problem.

*Supported by the "ex-60%" funds and by the biennial projects CPDA104492 and CPDA124755 of the University of Padova, and by the GNCS-INdAM.

[1] Dept. of Mathematics, University of Padova, Italy
e-mail: alvise,marcov@math.unipd.it

It is worth quoting the Hermitian Lanczos method for bivariate OPs with respect to a discrete inner product, studied in [22]. More recently, an interesting approach, based on an inverse eigenvalue problem to compute recurrence coefficients, has been proposed in [31] and tested on the square. On the other hand, a symbolic approach has been developed in [11], namely the MOPS library has been implemented for various computations involving classical multivariate OPs. In general, however, numerical codes devoted to multivariate OPs do not seem to be widely available.

In the present paper we pursue a basic orthogonalization approach, resorting to new algebraic cubature formulas based on product Gaussian quadrature on various nonstandard geometries, such as convex and concave polygons [26], circular sections (sectors, zones, lenses, lunes) [7]-[9]. We compute families of corresponding OPs by a two-step orthogonalization procedure, aimed at facing as much as possible ill-conditioning of the underlying Vandermonde-like matrices.

We provide a Matlab implementation of the method, and we make numerical tests on the quoted regions, in particular concerning discretized orthogonal projection of functions sampled at the cubature nodes, a method called "hyperinterpolation", originally introduced in the seminal paper by Sloan [25] as a good multivariate alternative to polynomial interpolation. Applications of positive cubature, multivariate OPs and hyperinterpolation arise, for example, in optical design [2], and geomathematics [20]. Our Matlab codes for positive cubature on nonstandard regions, multivariate OPs and hyperinterpolation are available at [5].

2 Computing multivariate OPs

In principle, the numerical computation of an OP basis could be done by the Gram-Schmidt process starting from any polynomial basis, as soon as the underlying inner product (with respect to some finite positive measure) is computable itself. In practice, however, severe instabilities can occur, depending strongly on the choice of the initial basis, which in turn determines the conditioning of the corresponding Vandermonde and Gram matrices. As known, ill conditioning is particularly severe, for example, dealing with the standard monomial basis.

Assume that a *positive* (i.e., with positive weights) cubature formula of polynomial degree of exactness (at least) $2n$ be available on a given compact domain (or manifold) $\Omega \subset \mathbb{R}^d$, and let us term $M = M_{2n}$ the cardinality of such a formula,

$$\int_{\Omega} f(\boldsymbol{x})\, d\mu = \sum_{i=1}^{M} w_i\, f(\boldsymbol{x}_i) \ , \ \ \forall f \in \mathbb{P}^d_{2n}(\Omega) \ , \qquad (1)$$

where $\mathbb{P}^d_m(\Omega)$ denotes the vector subspace of total degree polynomials in d

real variables of total degree not exceeding m, restricted to Ω, and $d\mu = \sigma(\boldsymbol{x})\,d\boldsymbol{x}$, is an absolutely continuous measure with density $\sigma \in L^1_+(\Omega)$.

Given a polynomial basis $\mathrm{span}(p_1, \dots, p_N) = \mathbb{P}^d_n(\Omega)$, consider the corresponding rectangular Vandermonde-like matrix

$$V_p = (v_{ij}) = (p_j(\boldsymbol{x}_i)) \in \mathbb{R}^{M \times N} \ , \ \ N = N_n = \dim(\mathbb{P}^d_n(\Omega)) \ , \tag{2}$$

and the diagonal $M \times M$ matrix

$$\sqrt{W} = \begin{pmatrix} \sqrt{w_1} & 0 & \dots & 0 \\ 0 & \sqrt{w_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{w_M} \end{pmatrix} . \tag{3}$$

Observe that $G_p = (g_{hk}) = (\sqrt{W}V_p)^t\sqrt{W}V_p$ is a positive definite Gram matrix for the inner product generated by the measure $d\mu$,

$$g_{hk} = \sum_{i=1}^{M} \sqrt{w_i}p_h(\boldsymbol{x}_i)\sqrt{w_i}p_k(\boldsymbol{x}_i) = \int_\Omega p_h(\boldsymbol{x})p_k(\boldsymbol{x})\,d\mu \ \ , 1 \le h, k \le N \ . \tag{4}$$

Then, the matrix $\sqrt{W}V_p$ and thus V_p are full-rank, and we can apply the QR factorization

$$\sqrt{W}V_p = QR \ , \ \ Q \in \mathbb{R}^{M \times N} \ , \ \ R \in \mathbb{R}^{N \times N} \ , \tag{5}$$

where Q is an orthogonal matrix and R an upper triangular nonsingular matrix. This means that the polynomial basis $(\phi_1, \dots, \phi_N) = (p_1, \dots, p_N)R^{-1}$ is *orthonormal*, since the corresponding Vandermonde-like matrix, say $V_\phi = V_pR^{-1}$, satisfies the property

$$G_\phi = (\sqrt{W}V_\phi)^t(\sqrt{W}V_\phi) = Q^tQ = I \ ,$$

that is, the Gram matrix of the new basis is the identity. Of course, the triangular matrix R^{-1} is nothing else than the coefficient matrix for a change of basis, corresponding to the Gram-Schmidt orthogonalization process with respect to the inner product in (4).

Working with finite precision arithmetic, the situation can be quite different. If the original Vandermonde-like matrix V_p is severely ill-conditioned, the matrix Q is not numerically orthogonal. However, when the conditioning (in the 2-norm) is below (or even not much greater) than the reciprocal of machine precision (rule of thumb), it is well known that a second QR factorization, say $Q = Q_1R_1$, succeeds in producing a matrix Q_1 orthogonal up to an error close to machine precision. This phenomenon is known as “twice is enough” in the framework of numerical orthogonalization, cf., e.g., [18]. A key point is that the second orthogonalization is applied to a matrix that, even though not orthogonal, is much better conditioned than the original one (under the conditions above).

Since we are not only interested in obtaining an orthogonal matrix, but also in determining the right change of basis, we apply the rule of "twice is enough" in the following way

$$\sqrt{W}V_\phi = \sqrt{W}V_pR^{-1} = Q_1R_1 \ , \tag{6}$$

where $V_\phi = V_pR^{-1}$ is well conditioned, so that the Vandermonde-like matrix

$$V_q = V_pR^{-1}R_1^{-1} \tag{7}$$

becomes nearly orthogonal, i.e.,

$$(q_1, \dots, q_N) = (p_1, \dots, p_N)R^{-1}R_1^{-1} \tag{8}$$

is a nearly orthonormal polynomial basis on K with respect to the inner product in (4). Here, "nearly orthogonal" means roughly that $\|G_q - I\|_2$ is extremely small (not too far from machine precision), where

$$G_q = (\sqrt{W}V_q)^t\sqrt{W}V_q \tag{9}$$

is the Gram matrix in the new basis.

Observe that if the original basis is "nested", i.e. $\text{span}(p_1, \dots, p_{N_s}) = \mathbb{P}_s^d(\Omega)$, $1 \le s \le n$, then the (nearly) orthonormal basis $(q_1, \dots, q_N)$ inherits the same property, since the matrices R and R_1 are upper triangular.

2.1 Hyperinterpolation

Once a (nearly) orthonormal basis $\{q_j\}$ is at hand, by the cubature formula we can also easily compute the discretized truncated orthogonal projection on $\mathbb{P}_n^d(\Omega)$ of a function $f \in L^2_{d\mu}(\Omega)$

$$\mathcal{L}_nf = \sum_{j=1}^{N} c_j\, q_j \ , \ \ c_j = \sum_{i=1}^{M} w_i\, q_j(\boldsymbol{x}_i)f(\boldsymbol{x}_i) \approx \int_\Omega q_j(\boldsymbol{x})f(\boldsymbol{x})\, d\mu \ . \tag{10}$$

Setting $\boldsymbol{c} = (c_1, \dots, c_N)^t$ and $\boldsymbol{f} = (f(\boldsymbol{x}_1, \dots, f(\boldsymbol{x}_M))^t$, the coefficients $\{c_j\}$ can be computed in vector form as

$$\boldsymbol{c} = (WV_q)^t\boldsymbol{f} = V_q^tW\boldsymbol{f} \ . \tag{11}$$

If the basis is only nearly orthonormal, this means that we are computing an approximate solution $\boldsymbol{c}$ to the system of the normal equations (cf. (9))

$$G_q\boldsymbol{c}^* = (WV_q)^t\boldsymbol{f} \ ,$$

or in other words that $\boldsymbol{c}^*$ corresponds, in view of (11), to solve a system with the identity matrix $A = I$ perturbed by the matrix $\delta A = G_q - I$. Standard estimates on the solution of systems with a perturbed matrix (cf., e.g., [29, §7.2]) give then immediately

$$\frac{\|\boldsymbol{c}-\boldsymbol{c}^*\|_2}{\|\boldsymbol{c}^*\|_2} \leq \|G_q - I\|_2 \;, \tag{12}$$

i.e., (11) gives the coefficients of the discretized orthogonal projection with a relative error bounded by the "distance from orthogonality" (which is not too far from machine precision).

In view of exactness of the formula at degree $2n$, the polynomial $\mathcal{L}_n f$ is a projection $L^2_{d\mu}(\Omega) \to \mathbb{P}^d_n(\Omega)$, called *hyperinterpolation*; indeed, it is the orthogonal projection with respect to the discrete inner product defined by the cubature formula. This notion was introduced and studied by Sloan in [25], where it is proved, for example, that for every $f \in C(\Omega)$

$$\|\mathcal{L}_n f - f\|_{L^2_{d\mu}(\Omega)} \leq 2\mu(K)\, E_n(f;\Omega) \to 0 \;, \;\; n \to \infty \;, \tag{13}$$

where $E_n(f;\Omega) = \inf\{\|f-p\|_\infty \,,\; p \in \mathbb{P}^d_n(\Omega)\}$.

Since then, multivariate hyperinterpolation has been successfully applied in various instances, as an interesting alternative to polynomial interpolation; cf., e.g., [1, 10, 19, 20].

A relevant quantity is also the uniform norm of $\mathcal{L}_n : C(\Omega) \to \mathbb{P}^d_n(\Omega)$, that is the operator norm with respect to $\|f\|_\Omega = \max_{\boldsymbol{x}\in\Omega} |f(\boldsymbol{x})|$. Observing that

$$\mathcal{L}_n f(\boldsymbol{x}) = \sum_{j=1}^{N} c_j\, q_j(\boldsymbol{x}) = \sum_{j=1}^{N} q_j(\boldsymbol{x}) \sum_{i=1}^{M} w_i\, q_j(\boldsymbol{x}_i) f(\boldsymbol{x}_i) = \sum_{i=1}^{M} f(\boldsymbol{x}_i)\psi_i(\boldsymbol{x}) \;,$$

where

$$\psi_i(\boldsymbol{x}) = K_n(\boldsymbol{x},\boldsymbol{x}_i) = w_i \sum_{j=1}^{N} q_j(\boldsymbol{x}) q_j(\boldsymbol{x}_i) \;,$$

and $K_n(\boldsymbol{x},\boldsymbol{y})$ is the reproducing kernel of $\mathbb{P}^d_n(\Omega)$ with the underlying inner product [13, Ch. 3], one can prove that

$$\|\mathcal{L}_n\| = \sup \frac{\|\mathcal{L}_n f\|_\Omega}{\|f\|_\Omega} = \max_{\boldsymbol{x}\in\Omega} \sum_{i=1}^{M} |\psi_i(\boldsymbol{x})| \;. \tag{14}$$

Theoretical or numerical bounds for (14) allow to estimate the hyperinterpolation convergence rate

$$\|\mathcal{L}_n f - f\|_\Omega \leq (1 + \|\mathcal{L}_n\|)\, E_n(f;\Omega) \;, \tag{15}$$

as well as to study the response of hyperinterpolation to perturbations of the sampled values, that is its stability.

In the numerical results section, we will show examples of hyperinterpolation on planar compact domains apparently not treated before in the numerical literature, for example polygons, and regions of the disk.

2.2 Implementation

We have implemented the orthogonalization procedure in the bivariate case, by a Matlab function named `multivop`. Such an implementation has been discussed also in [33]. As we have seen above, there are two main ingredients: the availability of a positive cubature formula on the compact domain, and the availability of a polynomial basis that is not too badly conditioned. Our standard general-purpose approach is to use the total-degree product Chebyshev basis of the smallest Cartesian rectangle containing the cubature nodes, with the graded lexicographical ordering (cf. [13]). Otherwise, the code accepts a user-provided basis, through the call to a function that computes the corresponding rectangular Vandermonde-like matrix at a given mesh of points. The call to such a function is of the type `V=myvand(deg,mymesh)`, where `deg` is the polynomial degree, and `mymesh` a 2-column or array of mesh points coordinates.

Since the first orthogonalization step has in practice the only purpose of constructing a better conditioned basis, to improve efficiency we apply the first QR factorization in (5) directly to the matrix V_p instead of $\sqrt{W}V_p$. This choice does not produce substantial differences from the numerical point of view, but avoids to compute the matrix product $\sqrt{W}V_p$, which corresponds to scaling the matrix rows by the square roots of the cubature weights. Such a scaling is performed only before the second orthogonalization step, by storing the matrix $\sqrt{W}$ in the Matlab `sparse` format.

Another key feature, in order to face ill-conditioning, is to compute the Vandermonde-like matrix in the orthonormal basis by the Matlab operation sequence $\mathtt{V}_q$`=(`$\mathtt{V}_p$`/R)/`$\mathtt{R}_1$; cf. [23] for the documentation on the Matlab / (`mrdivide`) operator. This is not equivalent, numerically, either to compute $\mathtt{V}_q$`=`$\mathtt{V}_p$`*inv(R)*inv(`$\mathtt{R}_1$`)`, or $\mathtt{V}_q$`=`$\mathtt{V}_p$`*inv(`$\mathtt{R}_1$`*R)`, or $\mathtt{V}_q$`=`$\mathtt{V}_p$`/(`$\mathtt{R}_1$`*R)`; indeed, all these three latter computations suffer from the possible ill-conditioning of `R` (inherited from $\mathtt{V}_p$), and do not ensure (near) orthogonality of `sqrt(W)*`$\mathtt{V}_q$.

A drawback is given by the necessity of producing in output (and store separately) both the upper triangular matrices `R` and $\mathtt{R}_1$, instead of the product $\mathtt{R}_1$`*R`, for any further application that requires the orthonormal basis.

The structure of the Matlab function `multivop` is sketched in Table 4. Such a function computes also the hyperinterpolation coefficients array, say `cfs`, of a function `fun` up to degree `deg`, cf. (10)-(11), as well as a numerical estimate of the uniform norm of the hyperinterpolation operator, cf. (14). Observe that the hyperinterpolant values at the cubature nodes are `hypval=`$\mathtt{V}_q$`'*cfs`. In general, given a Vandermonde-like matrix in the original basis on a given mesh, $\mathtt{U}_p$`=myvand(deg,mymesh)`, the values of the hyperinterpolant at that mesh can be computed as `hypval=`$\mathtt{U}_q$`'*cfs`, where again `R` and $\mathtt{R}_1$ are “inverted” separately, $\mathtt{U}_q$`=(`$\mathtt{U}_p$`/R)/`$\mathtt{R}_1$.

3 Numerical examples

A key feature of the orthogonalization method described in the previous section, is the availability of a positive cubature formula on the given compact domain exact up to polynomial degree $2n$, with as less nodes as possible in view of efficiency. Indeed, all the matrix operations involved, in particular the QR factorizations, have a computational complexity essentially proportional to the number of nodes (that is the number of rows of the Vandermonde-like matrices).

Unfortunately, *minimal* cubature formulas, i.e., formulas that achieve a given polynomial degree of exactness with a minimal number of nodes, are known in few multivariate instances, and mainly on standard geometries; cf., e.g., [6] and references therein. It is worth recalling, however, that in recent years some attention has been devoted to the computation of minimal or near minimal formulas on virtually arbitrary geometries by suitable optimization algorithms, which are also able to impose the positivity constraint; cf., e.g., [30, 32]. Cubature formulas with a given polynomial degree of exactness and cardinality equal to the dimension of the corresponding polynomial space, can be constructed by discrete extremal sets on various nonstandard geometries (e.g., polygons), but in general, though numerically stable, they present some negative weights; cf., e.g., [16], and [3] for the computation of multivariate extremal sets of Fekete and Leja type.

On the other hand, several positive cubature formulas based on product Gaussian quadrature have been obtained for the ordinary area or surface measure on nonstandard geometries, via suitable geometric transformations; cf., e.g., [7, 8, 17, 21, 27]. We present here some orthogonalization examples that exploit such product type formulas.

All the numerical tests have been made in Matlab 7.7.0 with an Athlon 64 X2 Dual Core 4400+ 2.40GHz processor. The Matlab codes are available online at [5].

3.1 Polygons

Orthogonal polynomials for the area measure on polygons seem to have been computed only in the regular case, see, e.g., [12] where group theory and rational exact arithmetics are used on the regular hexagon. In [26], a triangulation-free positive cubature formula for the standard area measure on a wide class of polygons was presented. The formula, implemented by the Matlab function `PolyGauss` [5], is based on product Gaussian quadrature by decomposition into trapezoidal panels, and works on all convex polygons, as well as on a class of nonconvex polygons with a simple geometric characterization [26, Rem. 4]. The cardinality for exactness degree n is bounded by $Ln^2/4 + \mathcal{O}(n)$, where L is the number of polygon sides.

Here, we apply such a formula to the computation of OPs on polygons by the function `multivop`. We consider two test polygons, see Figure 1-top, and five test functions with different degree of regularity and variation rate,

$$f_1(x,y) = (x+y+2)^{15}\ ,\ \ f_2(x,y) = \cos(x+y)\ ,\ \ f_3(x,y) = \cos(5(x+y))\ ,$$

$$f_4(x,y) = ((x-0.5)^2+(y-0.5)^2)^{3/2}\ ,\ \ f_5(x,y) = ((x-0.5)^2+(y-0.5)^2)^{5/2}\ . \tag{16}$$

Observe that f_1 is a polynomial, f_2 and f_3 are analytic functions, whereas $f_4 \in C^3$ and $f_5 \in C^5$, with the singular point $(0.5, 0.5)$ in the interior of both polygons (it is also in the interior of one of the trapezoidal panels, where the cubature nodes cluster more slowly than at the panel boundary). As polynomial basis, we have used the total-degree product Chebyshev basis of the smallest Cartesian rectangle containing the polygon, with the graded lexicographical ordering. The dimension of the polynomial spaces is $N = \dim(\mathbb{P}_n^2) = (n+1)(n+2)/2$.

It is worth stressing, see Figure 1-center, that while the matrix conditioning increases exponentially, the two-step orthogonalization procedure is able to get orthogonality up to an error close to machine precision.

The relative reconstruction error by hyperinterpolation is estimated as

$$\frac{\left(\sum_{i=1}^{M} w_i(\mathcal{L}_n f(\boldsymbol{x}_i) - f(\boldsymbol{x}_i))^2\right)^{1/2}}{\left(\sum_{i=1}^{M} w_i f^2(\boldsymbol{x}_i)\right)^{1/2}} \approx \frac{\|\mathcal{L}_n f - f\|_{L^2(\Omega)}}{\|f\|_{L^2(\Omega)}}\ . \tag{17}$$

Notice that, as expected since $\mathcal{L}_n$ is a projection operator, the error on the polynomial f_1 becomes close to machine precision as soon as exactness reaches the polynomial degree. The other errors are influenced by the smoothness level of the functions, as expected from (13) and the fact that a polygon is a “Jackson compact”, being a finite union of triangles.

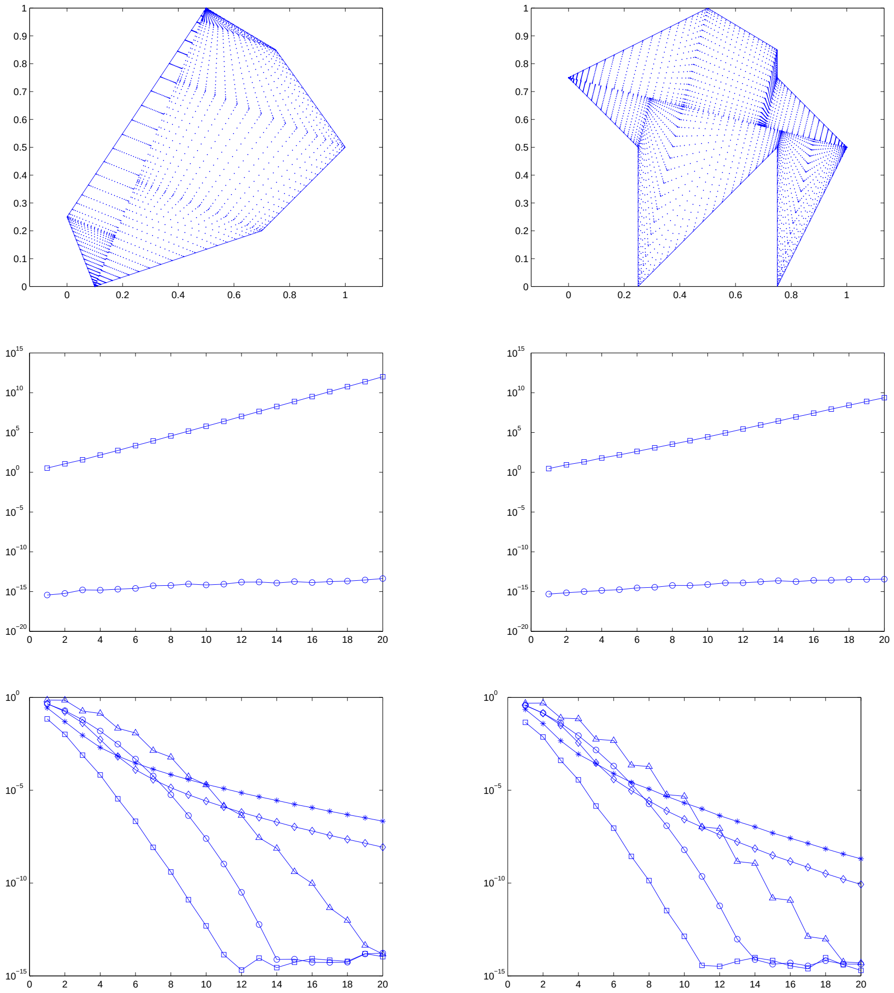
We recall that a compact $\Omega \subset \mathbb{R}^d$ which is the closure of an open bounded set, is termed a *Jackson compact* if it admits a *Jackson inequality*, namely for each $k \in \mathbb{N}$ there exist a positive integer m_k and a positive constant c_k such that

$$n^k\, E_n(f;\Omega) \le c_k \sum_{|\boldsymbol{i}| \le m_k} \|D^{\boldsymbol{i}} f\|_\Omega\ ,\ n > k\ ,\ \ \forall f \in C^{m_k}(\Omega)\ . \tag{18}$$

Examples of Jackson compacts are d-dimensional cubes (with $m_k = k+1$) and Euclidean balls (with $m_k = k$). We refer the reader to [24] for a recent survey on the multivariate Jackson inequality.

In Table 1, we report the number of nodes needed by `PolyGauss` to ensure exactness degree $2n$, together with the overall computing time of the function `multivop`, for some values of the polynomial degree n.

Figure 1: Top: cubature nodes for degree of exactness 40 on a 6-side convex (left) and a 9-side nonconvex (right) polygon; Center: condition number cond(V_p) (□) and orthogonality check $\|(WV_q)^t WV_q - I\|_2$ (○) for the sequence of degrees $n = 1, 2, \ldots, 20$; Bottom: reconstruction errors (17) by hyperinterpolation of the five test functions (16): f_1 (○), f_2 (□), f_3 (△), f_4 ($*$), f_5 (◊).



3.2 Circular sections

In [4, 8], trigonometric interpolation and quadrature on subintervals of the period have been studied; in particular, “subperiodic” Gaussian-type quadrature formulas have been implemented. In [15], the quadrature problem has been inserted in the more general framework of sub-range Jacobi

Table 1: Cardinality of the cubature formula (exactness degree $2n$) and computing time for the Matlab function `multivop` on the two test polygons of Figure 1.

n	5	10	15	20
convex polyg				
cardinality	252	792	1632	2772
cpu time	0.02s	0.17s	0.53s	1.67s
nonconvex polyg				
cardinality	378	1188	2448	4158
cpu time	0.03s	0.28s	0.88s	2.70s

polynomials. These results have opened the way to the generation of product Gaussian quadrature formulas for the area measure on several domains constructed by circular arcs, such as circular segments, sectors, zones, and more generally regions obtained by linear blending of circular/elliptical arcs. The basic idea is that such regions are image of a rectangle by a suitable injective analytic transformation, say $\boldsymbol{\sigma} : \mathcal{R} \to \Omega$, whose components belong to the tensor-product space $\mathbb{P}_1 \bigotimes \mathbb{T}_1$, and thus any polynomial $p \in \mathbb{P}_n^2(\Omega)$ corresponds to a tensorial algebraic-trigonometric polynomial $p \circ \sigma \in \mathbb{P}_n \bigotimes \mathbb{T}_n$ (where $\mathbb{T}_k$ denotes the space of trigonometric polynomials of degree not exceeding k, restricted to the relevant angular subinterval). By a change of integration variables

$$\int_\Omega p(\boldsymbol{x})\, d\boldsymbol{x} = \int_\mathcal{R} p(\boldsymbol{\sigma}(\boldsymbol{y}))|\mathrm{det}J\boldsymbol{\sigma}(\boldsymbol{y})|\, d\boldsymbol{y},$$

we can obtain a product formula, observing that $\mathrm{det}J\boldsymbol{\sigma} \in \mathbb{P}_1 \bigotimes \mathbb{T}_1$ and has constant sign. For details and examples we refer the reader to [7, 8].

The method has been also applied to regions related to a couple of overlapping disks with possibly different radii, such as lenses (intersection), double bubbles (union) and lunes (difference), cf. [8, 9]. The cardinality of such formulas for exactness degree n is $cn^2 + \mathcal{O}(n)$, with $c = 1$ or $c = 1/2$ (and even $c = 1/4$ by symmetry arguments in the special case of circular segments). All the corresponding Matlab codes are available at [5]. Concerning applications it is worth recalling, for example, that bivariate orthogonal polynomials and Gaussian quadrature on circular sections are useful tools in optical design, cf., e.g., [2].

In Figure 2, we show the same numerical tests of Figure 1, on a circular zone and on a circular lune. The dimension of the polynomial spaces is $N = \dim(\mathbb{P}_n^2) = (n + 1)(n + 2)/2$, and again we have chosen as starting basis the total-degree product Chebyshev basis of the smallest Cartesian rectangle containing the cubature nodes, with the graded lexicographical ordering. Observe that the singular point $(0.5, 0.5)$ for f_4 and f_5, belongs to the interior of both regions, where the cubature nodes cluster more slowly than at the boundary. We recall that circular zones and lunes, as well as all the arc-based regions quoted above, admit a multivariate Jackson inequality, cf. [7, 9].

In these tests, the hyperinterpolation errors on the less regular functions are higher than in the polygon examples. Such a behavior can be partially explained by the fact that the cubature formula uses much less points, and thus a less dense sampling, at the same degree of exactness (see Table 2).

In Figure 3 we report the uniform norm of the hyperinterpolation operator as a function of the degree n, for the four planar examples; cf. (14). Such norm is evaluated numerically by the Matlab function `multivop` via a simple matrix computation involving Vandermonde-like matrices in the (nearly) orthonormal basis (8), by a suitable control mesh. We have chosen as control mesh the cubature nodes for exactness degree $4n$, say $\{\boldsymbol{y}_1, \dots, \boldsymbol{y}_H\}$, thus computing, in view of (14),

$$\|\mathcal{L}_n\| \approx \max_{\boldsymbol{x} \in \{\boldsymbol{y}_h\}} \sum_{i=1}^{M} |\psi_i(\boldsymbol{x})| = \|W(V_p R^{-1} R_1^{-1})(U_p R^{-1} R_1^{-1})^t\|_1 ,$$

where V_p is defined in (2) and $U_p = (u_{hj}) = (p_j(\boldsymbol{y}_h)) \in \mathbb{R}^{H \times N}$. It is worth stressing that the increase rate experimentally observed, $\|\mathcal{L}_n\| = \mathcal{O}(n \log n)$, is not surprising at least concerning the disk sections, since it is compatible with known theoretical estimates for hyperinterpolation over the whole disk [19].

To conclude the planar examples, we increase the polynomial degree taking $n = 40$ in all the test regions. The numerical results are displayed in Table 3. In the polygon examples, we have extrapolated the condition numbers of the Chebyshev-Vandermonde matrices by linear regression in log scale, see Figures 1 and 2 bottom, since the `cond` function of Matlab underestimates the conditioning, giving values around the reciprocal of machine precision.

Despite the severe ill-conditioning, the results are still acceptable, in particular near-orthogonality occurs. There is a big increase of the computing time, explained by the fact that both, the number of cubature nodes as well as the dimension of the polynomial space, are roughly proportional to n^2.

3.3 Conclusions and future work

We have discussed a numerical method, based on standard numerical linear algebra, that computes orthogonal polynomials (OPs) as well as the corresponding hyperinterpolation operator with respect to an absolutely continuous measure on a planar region, provided that an algebraic cubature formula for the measure is known. The method is useful on nonstandard domains where an OP basis is not known analytically, such as for example general polygons and regions of the disk.

In the present implementation, it could work also on (compact subsets of) manifolds, as soon as a polynomial basis on the manifold is available, with a not too ill-conditioned Vandermonde-like matrix at the cubature nodes. For example, one might think applying the method to latitude-longitude rectangles of the sphere (with spherical caps as a special case, cf. [17]), using a basis of $(n+1)^2$ spherical harmonics. The latter, however, tends to be very ill-conditioned already at small degrees, since spherical harmonics are orthogonal and thus well-conditioned only on the whole sphere.

A further step to manage such situations, could be that of generating OPs that are tailored to a given discrete set, starting for example from the product Chebyshev basis of the smallest parallelepiped containg the set, and computing a suitable orthogonal polynomial basis by QR factorization with column-pivoting of the (scaled) Chebyshev-Vandermonde matrix, with possible rank-revealing (the rank being the dimension of the underlying polynomial space).

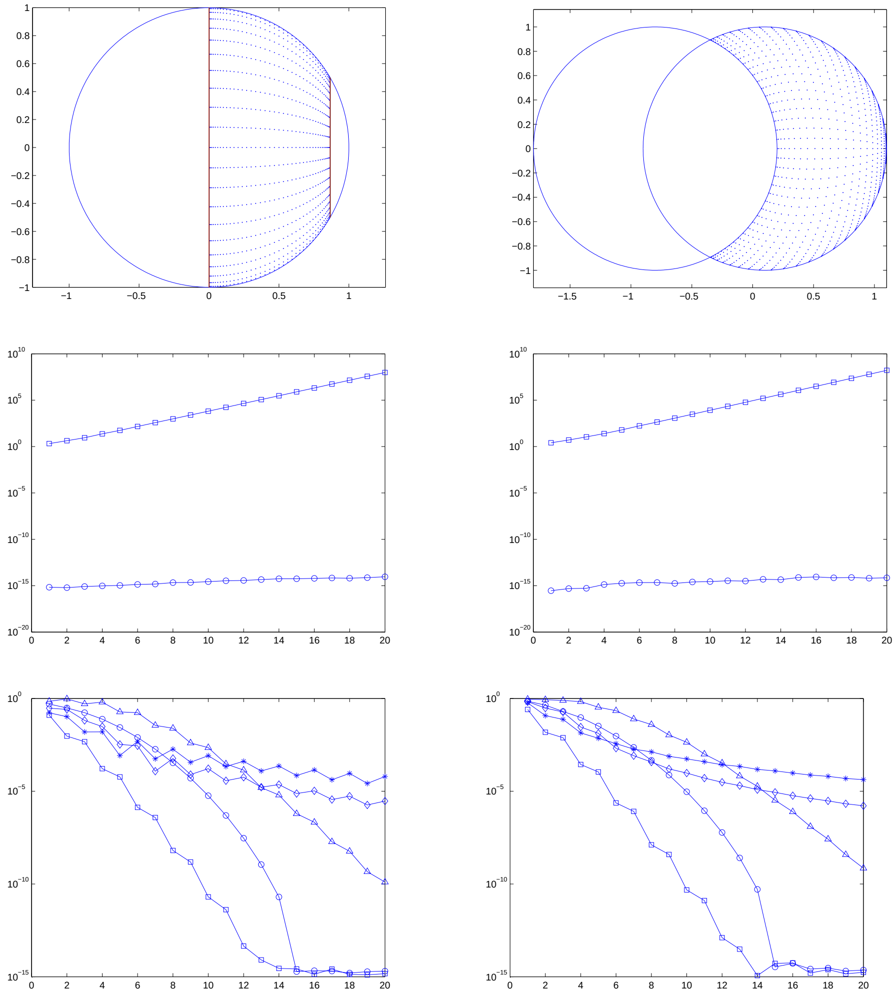
Another interesting direction, especially when we should individuate sampling nodes on a region to be used for the recovery of several functions, is that of obtaining low-cardinality algebraic cubature formulas by compression of the discrete measure corresponding to the original formula, following the approach discussed in [28].

Acknowledgements. The authors wish to thank Gerard Meurant for some helpful discussions.

Table 2: Cardinality of the cubature formula (exactness degree $2n$) and computing time for the Matlab function `multivop` on the two test regions of Figure 2.

n	5	10	15	20
circular zone				
cardinality	78	253	528	903
cpu time	0.01s	0.09s	0.30s	0.53s
circular lune				
cardinality	78	253	528	903
cpu time	0.01s	0.08s	0.33s	0.51s

Figure 2: Top: cubature nodes for degree of exactness 40 on a circular zone (left) and on a circular lune (right); Center and Bottom: as in Figure 1 for the zone and the lune.



References

[1] K. Atkinson, W. Han, Spherical Harmonics and Approximations on the Unit Sphere: An Introduction, Lecture Notes in Mathematics, 2044, Springer, Heidelberg, 2012.

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

[3] L. Bos, S. De Marchi, A. Sommariva, M. Vianello, Computing multivariate Fekete and Leja points by numerical linear algebra, SIAM J. Numer. Anal. 48

Figure 3: Numerically evaluated uniform norms of the hyperinterpolation operator for degree $n = 1, \dots, 20$ on the four planar examples above: convex polygon ($\triangle$), nonconvex polygon ($+$), circular zone ($\square$) and lune ($\circ$), compared with $\frac{1}{2} n \log n$ and $n \log n$ (lower and upper ($*$) curves).

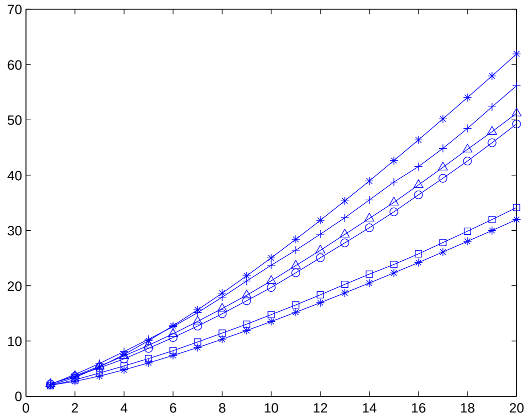


Table 3: Conditioning of the original Chebyshev-Vandermonde matrix, orthogonality check, cardinality of the cubature formula, hyperinterpolation error on f_4 and computing time for $N = 861$ orthonormal polynomials (degree $n = 40$) on the four regions above.

	cond	orth	card	f_4 err	cpu
convex polyg	1.4e+24	1.6e-13	10332	2.1e-10	50.6s
nonconvex polyg	1.1e+19	1.7e-13	15498	5.9e-14	78.2s
circular zone	1.4e+16	2.4e-14	3403	4.3e-06	18.5s
circular lune	3.4e+16	2.7e-14	3403	2.9e-06	17.9s

(2010), 1984–1999.

[4] L. Bos, M. Vianello, Subperiodic trigonometric interpolation and quadrature, Appl. Math. Comput. 218 (2012), 10630–10638.

[5] CAA Padova-Verona research group: numerical software, online at: `http://www.math.unipd.it/~marcov/CAAsoft.html`.

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

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

[8] G. Da Fies, M. Vianello, Trigonometric Gaussian quadrature on subintervals of the period, Electron. Trans. Numer. Anal. 39 (2012), 102–112.

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

[10] S. De Marchi, M. Vianello, Y. Xu, New cubature formulae and hyperinterpolation in three variables, BIT Numerical Mathematics 49 (2009), 55–73.

[11] I. Dumitriu, A. Edelman, G. Shuman, MOPS: multivariate orthogonal polynomials (symbolically), J. Symbolic Comput. 42 (2007), 587–620.

[12] C.F. Dunkl, Orthogonal polynomials on the hexagon, SIAM J. Appl. Math. 47 (1987), 343–351.

[13] C.F. Dunkl, Y. Xu, Orthogonal Polynomials of Several Variables, Encyclopedia of Mathematics and its Applications, 81, Cambridge University Press, Cambridge, 2001.

[14] W. Gautschi, Orthogonal Polynomials: Computation and Approximation, Oxford University Press, New York, 2004.

[15] W. Gautschi, Sub-range Jacobi polynomials, Numer. Algorithms 61 (2012), 649–657.

[16] M. Gentile, A. Sommariva, M. Vianello, Polynomial interpolation and cubature over polygons, J. Comput. Appl. Math. 235 (2011), 5232–5239.

[17] M. Gentile, A. Sommariva, M. Vianello, Polynomial approximation and quadrature on geographic rectangles, Appl. Math. Comput., 2016, to appear.

[18] L. Giraud, J. Langou, M. Rozloznik, J. van den Eshof, Rounding error analysis of the classical Gram-Schmidt orthogonalization process, Numer. Math. 101 (2005), 87–100.

[19] O. Hansen, K. Atkinson, D. Chien, On the norm of the hyperinterpolation operator on the unit disc and its use for the solution of the nonlinear Poisson equation, IMA J. Numer. Anal. 29 (2009), 257–283.

[20] K. Hesse, I.H. Sloan, Hyperinterpolation on the sphere, Frontiers in interpolation and approximation, 213–248, Pure Appl. Math. (Boca Raton), 282, Chapman & Hall/CRC, Boca Raton, FL, 2007.

[21] K. Hesse, R. Womersley, Numerical integration with polynomial exactness over a spherical cap, Adv. Comput. Math. 36 (2012), 451–483.

[22] M. Huhtanen, R.M. Larsen, On generating discrete orthogonal bivariate polynomials, BIT 42 (2002), 393–407.

[23] MathWorks, MATLAB Documentation Set, 2015 version, online at: `http://www.mathworks.com`.

[24] W. Pleśniak, Multivariate Jackson inequality, J. Comput. Appl. Math. 233 (2009), 815–820.

[25] I.H. Sloan, Interpolation and hyperinterpolation over general regions, J. Approx. Theory 83 (1995), 238–254.

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

[27] A. Sommariva, M. Vianello, Gauss-Green cubature and moment computation over arbitrary geometries, J. Comput. Appl. Math. 231 (2009), 886–896.

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

[29] G. Strang, Linear Algebra and its Applications. Academic, New York-London, 1976.

[30] M.A. Taylor, B.A. Wingate, L. Bos, A cardinal function algorithm for computing multivariate quadrature points, SIAM J. Numer. Anal. 45 (2007), 193–205.

[31] M. Van Barel, A. Chesnokov, A method to compute recurrence relation coefficients for bivariate orthogonal polynomials by unitary matrix transformations, Numer. Algorithms 55 (2010), 383–402.

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

[33] M. Zaccaron, Discrete orthogonal polynomials and hyperinterpolation over planar regions, Master Thesis in Mathematics, University of Padova, 2014 (advisor: M. Vianello).

Table 4: Matlab function for OPs and hyperinterpolation.

```
function [R,R1,cfs,hyperr,hypnorm] = multivop(deg,nodes,weights,controlmesh);

% Vandermonde in the polynomial basis on the cubature nodes
V = myvand(deg,nodes);
% scaling matrices
S = sparse(diag(sqrt(weights))); W = sparse(diag(weights));
% orthogonalization process
[Q,R] = qr(V,0);
V = V/R;
[Q,R1] = qr(S*V,0);
% Vandermonde in the OP basis on the cubature nodes
V = V/R1;
% sampling a function at the cubature nodes
f = myfun(nodes);
% hyperinterpolation coefficients of the function
cfs = V'*(weights.*f);
% estimating the hyperinterpolation error
hyperr = sqrt(weights'*(V*cfs-f).^2)/sqrt(weights'*f.^2);
% Vandermonde in the OP basis on the control nodes
U = (myvand(deg,controlmesh)/R)/R1;
% estimating the uniform norm of the hyperinterpolation operator
hypnorm = norm(W*V*U',1);
end
```