Compression of multivariate discrete measures and applications *

Alvise Sommariva and Marco Vianello†
Department of Mathematics, University of Padova (Italy)

July 11, 2016

Abstract

We discuss two methods for the compression of multivariate discrete measures, with applications to node reduction in numerical cubature and least-squares approximation. The methods are implemented in the Matlab computing environment, in dimension two.

2010 AMS subject classification: Primary 41A10, 65D32; Secondary 65D10.

Keywords: multivariate discrete measures, compression, algebraic cubature, node reduction, nonnegative least-squares (NNLS), QR factorization with column pivoting, Approximate Fekete Points, compressed least-squares.

1 Node reduction

Consider a multivariate discrete measure ν supported at a finite set $X = \{P_i\} \subset \mathbb{R}^d$, $i = 1, \ldots, M$, with correspondent weights (masses) $\boldsymbol{\omega} = \{\omega_i\}$ (to speak of a measure the weights should be positive, but the construction can be used also in general). "Compression" of the measure means that we try to compute the finite sum corresponding to integration of a function, by extracting a subset of the mass points and re-weighting them

$$S(f) = \int_{\mathbb{R}^d} f(P)\, d\nu = \sum_{i=1}^{M} \omega_i\, f(P_i) \approx \sum_{k=1}^{N} w_k\, f(Q_k) \tag{1}$$

where

$$\{Q_k\} \subset X\ ,\ N = \mathrm{card}(\{Q_k\}) \le \mathcal{V} = \dim(\mathbb{P}_n^d) < M\ ,$$

in such a way that the formula is exact, or nearly exact, on total-degree polynomials up to a degree n. Here and below, $\mathbb{P}_n^d$ denotes the space of d-variate total-degree polynomials of degree not exceeding n, with dimension $\mathcal{V} = \mathcal{V}(n,d) = \binom{n+d}{d}$.

*Work partially supported by the "ex-60%" funds and by the biennial project CPDA124755 of the University of Padova, and by the GNCS-INdAM.

†Corresponding author: e-mail: *marcov@math.unipd.it*

The discrete measure could typically correspond itself to a cubature formula for some absolutely continuous measure on a compact domain containing the points, possibly exact on polynomials of the same degree, but also to different situations, e.g. discrete least-squares approximation at the set X; see Sections 3-4 below.

As it well-known from cubature theory, the quality of approximation (1) depends mainly on two features: that $\sum |w_k|$ remains bounded, or at least increases very slowly with n, and how accurately the function can be (uniformly) approximated by polynomials of a given degree, say in some compact domain $K \supset X$. The first property, which ensures also stability, certainly holds when the weights are positive, since in this case $\sum |w_k| = \sum w_k$ is equal to $\sum \omega_i = \int 1 \, d\nu$. The second property can be related to multivariate Jackson-type results, which in turn depend on the geometry of K and on the regularity of f; cf., e.g., [30].

From the computational point of view, we should find an algorithm that extracts points and computes weights, such that

$$S(p) = \sum_{i=1}^{M} \omega_i \, p(P_i) = \sum_{k=1}^{N} w_k \, p(Q_k) \ , \quad \forall \, p \in \mathbb{P}_n^d \ . \tag{2}$$

A possible approach is to require moment matching on a total-degree polynomial basis, say $\{\phi_j\}$, $j = 1, \ldots, \mathcal{V}$, which gives the Vandermonde-like system in the unknowns $\boldsymbol{w}^* = \{w_i^*\}$

$$V^t \boldsymbol{w}^* = \boldsymbol{m} \ , \quad V = (v_{ij}) = (\phi_j(P_i)) \ , \quad \boldsymbol{m} = V^t \boldsymbol{\omega} \ , \tag{3}$$

$1 \le i \le M$, $1 \le j \le \mathcal{V}$, where $\boldsymbol{m} = \{m_j\}$ are the discrete moments of the basis $\{\phi_j\}$. Observe that such a system is underdetermined., and that V (and hence V^t) has full rank if and only if the discrete set X is $\mathbb{P}_n^d$-determining, i.e., a polynomial in $\mathbb{P}_n^d$ vanishing there is identically zero.

A first way to solve the system, is to ask that it be satisfied in the classical sense, by setting to zero $M - N$ components of $\boldsymbol{w}^*$. This can be conveniently done via the QR *algorithm with column pivoting* proposed in 1965 by Businger and Golub [6], which is implemented for example by the Matlab "backslash" operator. It is equivalent to a greedy selection of the columns in order to maximize the successive volumes, and eventually the (absolute value of the) determinant (the column selection problem being NP-hard, cf. [8]). If the matrix has full rank, the final result is a weight vector $\boldsymbol{w}^*$ where only $N \le \mathcal{V}$ components are nonzero, so that we can extract the points $\{Q_k\}$ from X by the column indexes corresponding to such components. A drawback is that the resulting weights $\boldsymbol{w} = \{w_k\}$ are not all positive, in general, but typically the negative ones are few and of small size, so that the relevant stability parameter

$$\sigma = \frac{\sum |w_k|}{|\sum w_k|} \ge 1 \tag{4}$$

is not far from 1.

We recall that this is exactly the approach adopted in [32] in the framework of multivariate polynomial interpolation, to extract the so-called "Approximate Fekete Points" from a given mesh of points of a compact set, in particular from a polynomial "admissible mesh" in the sense of Calvi and Levenberg [7].

This approach has been further developed for example in [3, 4], where it is also proved that (Fekete and) Approximate Fekete Points extracted from admissible meshes behave asymptotically like the true Fekete points, in the sense that the corresponding discrete probability measures with equal weights converge weakly to the pluripotential theoretic equilibrium measure of the compact set.

The use of Approximate Fekete Points in cubature formulas has already been explored for example in [21], but still in the framework of polynomial admissible meshes. Here we extract Approximate Fekete Points directly from the support of a discrete measure, for example from the nodes of an algebraic cubature formula as in Section 2, in order to reduce by re-weighting the number of nodes.

A second way to extract the points, which ensures positivity of the weights, is to solve the NNLS (*Non Negative Least Squares*) problem

$$\|\boldsymbol{m} - V^t \boldsymbol{w}^*\|_2 = \min \|\boldsymbol{m} - V^t \boldsymbol{u}\|_2 \ , \ \ \boldsymbol{u} \in \mathbb{R}^M \ , \ \ \boldsymbol{u} \geq \boldsymbol{0} \ . \tag{5}$$

This quadratic programming problem can be conveniently solved in Matlab by the `lsqnonneg` function, which is based on a variant of the active set optimization algorithm by Lawson and Hanson [25], and ensures sparsity of the solution $\boldsymbol{w}^*$. The residual of such a solution, say

$$\varepsilon_{mom} = \|\boldsymbol{m} - \boldsymbol{\mu}\|_2 \ , \ \ \boldsymbol{\mu} = \{\mu_j\} = V^t \boldsymbol{w}^* \ , \ \ \mu_j = \sum_{k=1}^{N} w_k \, \phi_j(Q_k) \ , \tag{6}$$

will not be zero, in general, but if it is sufficiently small we can speak of a "nearly exact" formula (whose nodes $\{Q_k\}$ and weights $\boldsymbol{w} = \{w_k\}$ are extracted correspondingly to the nonzero components of $\boldsymbol{w}^*$). Of course this does not mean that we expect the same error size in the integration of any polynomial, as it will be clear from the error analysis developed below, cf. (15)-(16).

This kind of approach for positive formulas is of course not new in numerical quadrature. It dates back at least to the 50's and 60's, cf., e.g., [15, 37] and the survey [9] with references therein. Recently, it has been successfully adopted for example in [24], where univariate instances are considered and Matlab `lsqnonneg` is used in the implementation. Here we apply the method systematically for the first time to the compression of general multivariate discrete measures.

Both the approaches sketched above suffer from ill-conditioning of the matrix V, so the use of a well-conditioned polynomial basis, especially at increasing exactness degrees, becomes mandatory in practice. Finding well-conditioned polynomial bases is still a challenging problem, in multivariate instances. It is well-known that the standard monomial basis becomes rapidly extremely ill-conditioned. Better results are obtained with bases orthogonal with respect to some measure, for example the total-degree Chebyshev basis of a rectangle containing the computational domain, but still the conditiong of the Vandermonde matrix becomes intractable already at moderate degrees. We recall that the total-degree Chebyshev orthogonal basis of a rectangle $[a_1, b_1] \times \cdots \times [a_d, b_d] \supset X$ is (cf. [17])

$$\left\{ \prod_{i=1}^{d} T_{h_i}(t_i) \right\} \ , \ \ t_i = t_i(x_i) = \frac{2x_i - b_i - a_i}{b_i - a_i} \ , \ \ h_i \geq 0 \ , \ \ 0 \leq \sum_{i=1}^{d} h_i \leq n \ , \tag{7}$$

where $T_h(\cdot) = \cos(h \arccos(\cdot))$ is the Chebyshev polynomial of the first kind of degree h.

A possible approach to get a well-conditioned basis is to orthogonalize with respect to a discrete measure, computing discrete orthogonal polynomials. In our case, it makes sense to orthogonalize with respect to a discrete measure with the same support X of ν, for example the measure with unit mass at each point of X. Some clever numerical linear algebra algorithms have been proposed for discrete orthogonalization of the bivariate monomial basis, e.g. in [23] and more recently in [36]. However, they become slow already at moderate degrees when the support cardinality is high, and in addition, [23] can suffer of loss of orthogonality (and thus needs some kind of re-orthogonalization), so their use should be probably reserved to situations where no alternative is possible.

In this paper, we start from the total-degree Chebyshev orthogonal basis of a rectangle (parallelepiped) containing the ($\mathbb{P}_n^d$-determining) discrete set X, and manage ill-conditioning by a standard QR orthogonalization of the multivariate Chebyshev-Vandermonde matrix. If

$$\boldsymbol{\tau}(P) = (\tau_1(P), \ldots, \tau_{\mathcal{V}}(P)) \tag{8}$$

is the suitably ordered (for example lexicographically) total-degree Chebyshev basis of the rectangle (in a row vector notation) and V its Vandermonde matrix at the discrete set X, $V = (\tau_j(P_i)) \in \mathbb{R}^{M \times \mathcal{V}}$, we compute the QR factorization

$$V = QR\,, \quad Q \in \mathbb{R}^{M \times \mathcal{V}}\,, \quad R \in \mathbb{R}^{\mathcal{V} \times \mathcal{V}}\,, \tag{9}$$

and replace the matrix V with Q in (3)-(5), which corresponds to work with the discrete orthogonal basis

$$\boldsymbol{\psi}(P) = \boldsymbol{\tau}(P) R^{-1}\,. \tag{10}$$

In contrast to interpolation processes where a similar approach has been used, (cf., e.g., [4, 32]), there is no need to compute the new orthogonal basis out of X, so there is no need to compute and store the inverse matrix R^{-1} (observe that R retains the conditioning of V). In practice, due to increasing ill-conditioning of V with the degree, the matrix Q (and thus the basis $\boldsymbol{\psi}$) might not be orthogonal, but in any case is much better conditioned than V. If orthogonality is seeked, a second QR factorization $Q = Q_1 R_1$ suffices to get orthogonality up to machine precision, a phenomenon known as "twice is enough" in numerical Gram-Schmidt orthogonalization, cf., e.g., [22]. Such an approach works until the conditioning of V is not too far from the reciprocal of machine precision, and in our numerical experiments allows to reach moderate exactness degrees (order of the tens) with an acceptable computing time, as it is shown in the next Sections.

The core of the Matlab implementation of the two compression algorithms in the bivariate case is given by the Matlab functions in the appendix. In the next Sections, we first give some error estimates, showing in particular the effect of the moment error ε_{mom}. Then, we discuss two applications of the compression procedures just described, with several examples.

The first one concerns node reduction of algebraic cubature formulas, where we consider recently proposed formulas for integration over polygons, and over circular sections. An interesting situation is when we have a finite disjoint (even disconnected) union of domains, where we know an algebraic cubature formula

on each. We stress that our approach to node reduction is based on the underdetermined linear system (3), where the unknowns are the weights. Nonlinear node reduction techniques have been explored in numerical quadrature, which compute minimal or near minimal formulas by considering also the nodes as unknowns of a nonlinear system. These techniques are usually based on a clever use of Newton's method, have some heuristic features, and typically require the use of orthogonal polynomial bases, together with good low cardinality initial guesses to reduce the computational effort; see, e.g., [18, 28, 29, 35, 38] and references therein. From this point of view, both the algorithms discussed in the present paper can be seen as an initial step for such more sophisticated node reduction methods. Indeed, being implemented by basic Matlab routines, they provide low cardinality cubature formulas with a reasonable computational cost.

The second application is devoted to compression of standard polynomial least-squares on suitable meshes, in order to obtain a weighted least-squares method with comparable error estimates, and a great reduction of the number of sampling points.

All the numerical tests have been made in Matlab 7.7.0 with an Athlon 64 X2 Dual Core 4400+ 2.40GHz processor. The relevant Matlab codes are available at [11].

2 Error estimates

Computing the weights by solving (3)-(5) in an orthogonal polynomial basis with respect to some measure λ supported in a compact set $\Omega \supseteq X$, allows also to estimate the effect of the moment error in integrating a function, at least when the integrand f is defined on the whole Ω. For this analysis, the measures ν supported in $X \subseteq \Omega$ and λ supported in Ω could be even arbitrary. We term $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_N)$ such an orthogonal basis in $d\lambda$, for convenience, having in mind the Chebyshev case $\boldsymbol{\phi} = \boldsymbol{\tau}$ with $\Omega = [a_1, b_1] \times \cdots \times [a_d, b_d] \supset X$ and $d\lambda$ the product Chebyshev measure of the rectangle

$$d\lambda = \frac{2^d}{\prod_{i=1}^d (b_i - a_i)} \, \frac{1}{\prod_{i=1}^d \sqrt{1 - t_i^2}} \, dP \, ,$$

cf. (7). On the other hand, we can consider also the fully discrete case $\Omega = X$, $\lambda = \nu$ and $\boldsymbol{\phi} = \boldsymbol{\psi}$.

First, observe that

$$\int_{\mathbb{R}^d} p(P) \, d\nu = \int_{\Omega} p(P) \, d\nu = \int_X p(P) \, d\nu = \langle \boldsymbol{c}, \boldsymbol{m} \rangle \, , \quad \forall p \in \mathbb{P}_n^d \, , \tag{11}$$

$\boldsymbol{c} = \{c_j\}$, $\boldsymbol{m} = \{m_j\}$, $1 \le j \le \mathcal{V}$, where

$$c_j = \frac{\int_\Omega p(P) \phi_j(P) \, d\lambda}{\left(\int_\Omega \phi_j^2(P) \, d\lambda \right)^{1/2}} \, , \quad m_j = \int_\Omega \phi_j(P) \, d\nu = \int_X \phi_j(P) \, d\nu \, ,$$

are the Fourier coefficients of p in the $d\lambda$-orthogonal basis $\boldsymbol{\phi}$ and the $d\nu$-moments of $\boldsymbol{\phi}$, respectively. Moreover,

$$\sum_{k=1}^N w_k p(Q_k) = \langle \boldsymbol{c}, \boldsymbol{\mu} \rangle \, , \tag{12}$$

where $\boldsymbol{\mu}$ are the approximate moments, the moment error being estimated by ε_{mom}, cf. (6). Then we can write the "*near exactness*" estimate

$$\left|\int_\Omega p(P)\,d\nu - \sum_{k=1}^N w_k p(Q_k)\right| = |\langle \boldsymbol{c}, \boldsymbol{m}-\boldsymbol{\mu}\rangle| \\ \leq \|\boldsymbol{c}\|_2\,\|\boldsymbol{m}-\boldsymbol{\mu}\|_2 = \|p\|_{L^2_{d\lambda}(\Omega)}\,\varepsilon_{mom}\,. \tag{13}$$

Now, take $f \in C(\Omega)$, and define

$$E_n(f;\Omega) = \min_{p\in\mathbb{P}_n^d} \|f-p\|_{L^\infty(\Omega)} = \|f-p_n^*\|_{L^\infty(\Omega)}\,, \tag{14}$$

i.e., p_n^* is the best uniform approximation polynomial of degree not greater than n for f on Ω. By a classical chain of inequalities in quadrature theory and (13) we get

$$\left|\int_\Omega f(P)\,d\nu - \sum_{k=1}^N w_k f(Q_k)\right| \leq \left|\int_\Omega f(P)\,d\nu - \int_\Omega p_n^*(P)\,d\nu\right| \\ + \left|\int_\Omega p_n^*(P)\,d\nu - \sum_{k=1}^N w_k p_n^*(Q_k)\right| + \left|\sum_{k=1}^N w_k p_n^*(Q_k) - \sum_{k=1}^N w_k f(Q_k)\right| \\ \leq \left(\nu(\Omega) + \sum_{k=1}^N |w_k|\right) E_n(f;\Omega) + \|p_n^*\|_{L^2_{d\lambda}(\Omega)}\,\varepsilon_{mom}\,.$$

Using the inequality

$$\|p_n^*\|_{L^2_{d\lambda}(\Omega)} \leq \|p_n^*-f\|_{L^2_{d\lambda}(\Omega)} + \|f\|_{L^2_{d\lambda}(\Omega)} \leq \sqrt{\lambda(\Omega)}\,\|f-p_n^*\|_{L^\infty(\Omega)} + \|f\|_{L^2_{d\lambda}(\Omega)}\,,$$

we obtain the final cubature error estimate

$$\left|\int_\Omega f(P)\,d\nu - \sum_{k=1}^N w_k f(Q_k)\right| \leq C\,E_n(f;\Omega) + \|f\|_{L^2_{d\lambda}(\Omega)}\,\varepsilon_{mom}\,,\quad \forall f\in C(\Omega)\,, \tag{15}$$

with

$$C = \nu(\Omega) + \sum_{k=1}^N |w_k| + \sqrt{\lambda(\Omega)}\,\varepsilon_{mom} = \nu(\Omega) + \sigma\left|\sum_{k=1}^N w_k\right| + \sqrt{\lambda(\Omega)}\,\varepsilon_{mom} \\ \leq \nu(\Omega) + \sigma\left(\nu(\Omega) + \left|\sum_{k=1}^N w_k - \int_\Omega 1\,d\nu\right|\right) + \sqrt{\lambda(\Omega)}\,\varepsilon_{mom} \\ \leq \nu(\Omega) + \sigma\left(\nu(\Omega) + \sqrt{\lambda(\Omega)}\,\varepsilon_{mom}\right) + \sqrt{\lambda(\Omega)}\,\varepsilon_{mom} = (1+\sigma)\left(\nu(\Omega) + \sqrt{\lambda(\Omega)}\,\varepsilon_{mom}\right)$$

where (13) has been used with $p\equiv 1$ and the roles of the moment error ε_{mom} and of the stability parameter σ are explicit, cf. (6) and (4). Observe that (15) englobes (13) since $E_n(f;\Omega)=0$ for $f\in\mathbb{P}_n^d$. Moreover, dividing the right-hand side of (15) by the $d\nu$-integral of f when it does not vanish, we see that

$$E_{mom}(f) = \alpha(f)\,\varepsilon_{mom}\,,\quad \alpha(f) = \frac{\left(\int_\Omega f^2(P)\,d\lambda\right)^{1/2}}{\left|\int_\Omega f(P)\,d\nu\right|}\,,\quad \int_\Omega f(P)\,d\nu \neq 0\,, \tag{16}$$

gives an upper bound for the moment error effect in a relative cubature error estimate.

3 Compressed algebraic formulas

Consider an algebraic cubature formula with positive weights on a domain $K \subset \mathbb{R}^d$

$$\int_X p(P)\, d\nu = \sum_{i=1}^{M} \omega_i\, p(P_i) = I(p) = \int_K p(P)\, \omega(P) dP\ ,\ \ \forall p \in \mathbb{P}_n^d\ , \qquad (17)$$

with respect to some absolutely continuous measure with density $\omega(P)$, where $X \subset K$, M and ν (i.e., the weights) depend on n, $M > \mathcal{V} = \dim(\mathbb{P}_n^d)$. We can try to reduce the cardinality of such a formula, by extracting nodes and reweighting as in (3) or (5). Below we present some examples in dimension $d = 2$ with $\omega(P) \equiv 1$, related to recently developed cubature formulas on nonstandard domains.

3.1 Circular sections

In some recent papers, algebraic cubature formulas have been obtained for several domains related to circular arcs, such as circular segments, sectors, zones, lenses, lunes; cf. [10, 13, 14]. These formulas are of product Gaussian type, and are based on "subperiodic" trigonometric approximation (cf. [5]), namely new trigonometric Gaussian quadrature formulas on subintervals of the period [12], like

$$\int_{\alpha_1}^{\alpha_2} t(\theta)\, d\theta = \sum_{j=1}^{n+1} \lambda_j\, t(\theta_j)\ ,\ \ \forall t \in \mathbb{T}_n\ ,$$

where $\mathbb{T}_n^1$ denotes the $2n+1$-dimensional space of univariate trigonometric polynomials of degree not greater than n. The angular nodes $\{\theta_j\}$ are obtained by a nonlinear transformation of algebraic Gaussian nodes for a suitable measure on $[-1, 1]$, and the weights $\{\lambda_j\}$ are positive.

The construction of the product Gaussian formulas exact on bivariate polynomials relies on the observation that many circular sections, say K, are suitable blending transformations $K = T(\mathcal{R})$, $\mathcal{R} = [0, 1] \times [\alpha_1, \alpha_2]$, of the form

$$T(s, \theta) = sP_1(\theta) + (1 - s)P_2(\theta)\ ,\ \ s \in [0, 1]\ ,\ \ \theta \in [\alpha_1, \alpha_2]\ ,$$

with $P_1(\theta)$ and $P_2(\theta)$ first degree trigonometric polynomials with vector coefficients (linear trigonometric arcs), and constant sign Jacobian $J(s, \theta) \in \mathbb{P}_1^1 \bigotimes \mathbb{T}_2^1$ (i.e. the transformation is injective). Then, for any total-degree bivariate polynomial $p \in \mathbb{P}_n^2$ we have that $p \circ T \in \mathbb{P}_n^1 \bigotimes \mathbb{T}_n^1$ and $(p \circ T)\, |J| \in \mathbb{P}_{n+1}^1 \bigotimes \mathbb{T}_{n+2}^1$, so that

$$\int_K p(P)\ dP = \int_0^1 \int_{\alpha_1}^{\alpha_2} p(T(s, \theta))\, |J(s, \theta)|\, ds d\theta$$

$$= \sum_{j=1}^{n+3} \sum_{h=1}^{\lceil (n+2)/2 \rceil} W_{hj}\, p(T(s_h, \theta_j))\ ,\ \ W_{hj} = \gamma_h \lambda_j |J(s_h, \theta_j)|\ , \qquad (18)$$

where $\{s_h\}$ and $\{\gamma_h\}$ are the Gauss-Legendre nodes and weights for $[0, 1]$, cf. [19, 20].

Then we have a formula of the form (17) with $\{\omega_i\} = \{W_{hj}\}$ and $\{P_i\} = \{T(s_h, \theta_j)\}$ (by a suitable bi-index ordering), and $M = \lceil (n + 3) \lceil (n + 2)/2 \rceil$.

In some situations M could be slightly smaller, namely when the Jacobian is independent of s or θ, but still greater than $\mathcal{V} = \dim(\mathbb{P}_n^2) = (n+1)(n+2)/2$; we refer the reader to [10] for a full discussion and several examples of such blending formulas.

Now, we can reduce the number of nodes by compression and re-weighting as in as in (3) or (5), the compression ratio being $N/M \lesssim (n+1)/(n+3)$. A first example appears in Figure 1 and Tables 1-2, concerning a circular zone of the unit disk (the region cut by two parallel lines) with angular interval $[\pi/6, \pi/2]$. In this case, a suitable transformation is $T(s,\theta) = (s\cos\theta, 1 + 2s\sin\theta)$ with $J(s,\theta) = 2s$. We report all the relevant parameters, including the coefficient α in (16), rounded to the displayed digits: notice that E_{mom} is a large overestimate of the actual effect of the moment error. Moreover, the value of α stabilizes, as it is expected since

$$\lim_{n\to\infty} \alpha(f) = \lim_{n\to\infty} \frac{\left(\int_\Omega f^2(P)\, d\lambda\right)^{1/2}}{\left|\sum_{i=1}^M \omega_i\, f(P_i)\right|} = \frac{\left(\int_\Omega f^2(P)\, d\lambda\right)^{1/2}}{\left|\int_K f(P)\, dP\right|} \,,$$

by positivity of the weights $\{\omega_i\}$ and exactness in $\mathbb{P}_n^d$, which ensure convergence of the denominator (the numerator has been computed by a high-degree Gauss-Chebyshev product formula, Ω being the smallest rectangle containing the domain).

In Table 2 we show the behavior of the compressed formulas on three test functions with different degree of regularity, a polynomial, a Gaussian, and a C^2 function with a singularity of the third derivatives

$$f_1(x,y) = (x+y)^n \,, \quad f_2(x,y) = e^{-((x-x_0)^2+(y-y_0)^2)} \,,$$

$$f_3(x,y) = ((x-x_0)^2 + (y-y_0)^2))^{3/2} \,, \quad (x_0,y_0) = (0.5, 0.5) \,. \tag{19}$$

The reference values of the integrals $I(f_1)$ are computed at machine precision by the original exact blending formula, whereas the reference values of the other two integrals, computed adaptively by the blending formula, are $I(f_2) = 0.9461023217911515$ rounded to 16 digits and $I(f_3) = 0.88384114$ rounded to 8 digits.

Here, and in all the examples below, the singularity (x_0, y_0) for f_3 lies inside the integration domain and is bounded away from the boundary (where the cubature nodes cluster). We stress that in this example, and also in all the ones below, the behavior of compression by QR with column pivoting (QRpiv) is quite satisfactory. The moment error is extremely small, the weights are not all positive but the stability parameter (4) remains close to 1, and the computing time is significantly lower than that of NNLS-based compression (which, however, guarantees positivity).

As a second example, with a more consistent node reduction, we consider special circular sections arising in the framework of optical design. Recently, Bauman and Xiao [1] have introduced quadrature methods based on prolate spheroidal wave functions, to treat for instance optical apertures (pupils) that are obscured and vignetted (a feature that occurs in optical astronomy). We consider 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 [1]; see Figure 2.

The resulting domain can be split into a standard annular sector (the unvignetted portion) and a vignetted region that cannot be treated directly by a blending formula (18), since the arcs correspond to different angular intervals. Such a region, however, can be conveniently split for example into three generalized sectors, as shown in Figure 2. Collecting together nodes and weights of the corresponding blending formulas, we obtain a composite cubature formula of polynomial degree of exactness n on the obscured and vignetted pupil, with $4(n^2/2 + \mathcal{O}(n)) = 2n^2 + \mathcal{O}(n)$ nodes and positive weights. Our compressed formulas for the pupil have then a compression ratio $N/M \lesssim 1/4$.

The numerical results appear in Tables 3 and 4. In this example $I(f_2) = 0.8508514261060272$ rounded to 16 digits and $I(f_3) = 3.2003163$ rounded to 8 digits. We note that the compressed formulas on the obscured and vignetted pupil seems even slightly more efficient with respect to the number of sampling points, than the formulas proposed in [1]. Indeed, for $n = 13$ the results in [1, §2.4] give a set of 112 nodes, whereas we have $N = \dim(\mathbb{P}^2_{13}) = 105$ nodes. On the other hand, the numerical results in Table 3 show high accuracy of the compressed formulas, with a moment error not exceeding 10^{-14} up to degree $n = 30$.

3.2 Polygons

In the recent literature on cubature methods, some attention has been devoted to integration over polygons; see, e.g., [21, 26, 31, 34, 38]. Indeed, despite the importance of polygons (and polyhedra) in computational geometry as well as in a wide range of applications, there was a gap in the numerical literature concerning cubature formulas tailored on polygons, emerged in particular in connection with the recent field of polygonal finite elements [28].

For the purpose of illustration, we apply the compression algorithms to the cubature formula proposed in [31]. As already observed, the present compression procedure can be seen as an initial step for more sophisticated node reduction methods, for example that proposed in [38].

The cubature formula of [31] is based on Green's integration theorem (the divergence theorem) and product Gaussian quadrature, and works by implicit trapezoidal panelization, avoiding explicit a priori partition of the polygonal domain into triangles or quadrangles. It applies to convex polygons, and more generally to a class of polygons (even nonconvex) with the following geometric property: there exists a special line (say ℓ), termed reference line, 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 $\ell \cap q$. Clearly such class contains all convex polygons, for example by taking the line connecting a pair of vertices with maximal distance. Observe that necessarily the reference line can intersect a polygon side only at a vertex.

The aspect of such formulas is the following. Given a polygon K with vertices $V_h = (\alpha_h, \beta_h)$, $h = 1, \ldots, L$ (counterclockwise ordered), and a suitable reference line, that up to a rotation we can identify with a line $x = \xi$ parallel to the y-axis, we have

$$\int_K p(P)\, dP = \sum_{h=1}^{L} \sum_{j=1}^{n_h} \sum_{l=1}^{\lceil n/2 \rceil} W_{hjl}\, f(\xi_{hjl}, \eta_{hj}) \;, \quad \forall p \in \mathbb{P}^d_n \;, \tag{20}$$

where $n_h = \lceil n/2 \rceil$ if $\Delta\alpha_h = 0$ or $n_h = \lceil n/2 \rceil + 1$ if $\Delta\alpha_h \neq 0$, and the nodes and weights are given by

$$\xi_{hjl} = \frac{x_h(s_j^{n_h}) - \xi}{2}\, s_l^{\lceil n/2 \rceil} + \frac{x_i(s_j^{n_h}) + \xi}{2}\ , \quad x_h(s) = \frac{\Delta\alpha_h}{2}\, s + \frac{\alpha_h + \alpha_{h+1}}{2}\ , \tag{21}$$

$$\eta_{hj} = y_h(s_j^{n_h})\ , \quad y_h(s) = \frac{\Delta\beta_h}{2}\, s + \frac{\beta_h + \beta_{h+1}}{2}\ , \tag{22}$$

$$W_{hjl} = \frac{1}{4}\, \Delta\beta_h\, \left(x_h(s_j^{n_h}) - \xi\right)\, \gamma_j^{n_h}\, \gamma_l^{\lceil n/2 \rceil}\ , \tag{23}$$

Δ denoting the usual forward difference operator, and $\{s_j^\mu\}$, $\{\gamma_j^\mu\}$ the nodes and weights of the Gauss-Legendre quadrature formula of degree of exactness μ on $[-1, 1]$, cf. [19]. By a suitable multi-index ordering, (20) is of the form (17) with $\{\omega_i\} = \{W_{hjl}\}$ and $\{P_i\} = \{(\xi_{hjl}, \eta_{hj})\}$. It is easily checked that the weights are all nonnegative, in view of the reference line properties and counterclockwise ordering of the vertices.

We present two examples, where we apply the compressed formulas to integration of the three test function (19) over the nonregular convex hexagon of Figure 3, and over the nonconvex nonagon of Figure 4. The numerical results are collected in Tables 5-8. Since $M \geq L\lceil n/2 \rceil^2$ and $N \leq \mathcal{V} = (n+1)(n+2)/2$, we expect a compression ratio $N/M \lesssim 2/L$. The reference values of the integrals are $I(f_2) = 0.4850601470247102$ rounded to 16 digits and $I(f_3) = 0.020571741$ rounded to 8 digits for the hexagon, and $I(f_2) = 0.4374093366938109$ and $I(f_3) = 0.017301869$ for the nonagon (again, the values of $I(f_1)$ are computed at machine precision by the original exact cubature formula).

Remark 1 (disconnected union). It is worth observing that nothing prevents using the compressed formulas on domains that are disconnected finite union of compact subregions. If an algebraic cubature formula is known on each subregion, we can collect together nodes and weights to get a "composite" cubature formula on the whole union. In order to reduce the number of nodes, we can then apply the compression algorithms, obtaining a formula for the union which much fewer nodes than before on each subregion. An example of this type is given in Figure 5 and Tables 9-10, where the union of six disconnected disks of different radius is concerned. A new phenomenon here appears. While the QR-piv algorithm gives results similar to those of the previous examples, the NNLS approach shows at degree 25 a moment error much larger than the others, of the order of 10^{-8}. This occurrence of course affects the integration errors of Table 10, even though, as already observed numerically, the moment error effect is much lower than E_{mom} in (16).

4 Compressed least-squares

Consider the classical discrete least-squares polynomial approximation in $\mathbb{P}_n^d$ of a function f at a discrete set $X \subset \mathbb{R}^d$,

$$X = \{P_i\}_{1 \leq i \leq M}\ , \quad M = \mathrm{card}(X)\ , \tag{24}$$

where X is $\mathbb{P}_n^d$-determining (or equivalently the Vandermonde like matrices corresponding to the polynomial space $\mathbb{P}_n^d$ have full rank). The discrete least-squares polynomial approximation is an orthogonal projection $L_n f$ on the polynomial space, with respect to the scalar product induced by the discrete measure ν with unit mass at each point

$$\|f - L_n f\|_{\ell^2(X)} = \min_{p \in \mathbb{P}_n^d} \|f - p\|_{\ell^2(X)} \,. \tag{25}$$

The discrete error of such an approximation is easily estimated

$$\begin{aligned} \|f - L_n f\|_{\ell^2(X)} &\leq \|f - p_n^*\|_{\ell^2(X)} \leq \sqrt{M}\, \|f - p_n^*\|_{\ell^\infty(X)} \\ &\leq \sqrt{M}\, \|f - p_n^*\|_{L^\infty(K)} = \sqrt{M}\, E_n(f;K) \;, \quad K \supset X \;, \end{aligned} \tag{26}$$

for $f \in C(K)$, where p_n^* is the best uniform approximation polynomial for f on a compact domain K containing X (for example K could be the closure of the convex hull of the points). If K is a Jackson compact, (26) estimates the approximation quality through the regularity of f, cf. [30].

If $M > \mathcal{V}(2n,d) = \dim(\mathbb{P}_{2n}^d)$, and X is $\mathbb{P}_{2n}^d$-determining, we can approximate the scalar product in $\ell^2(X)$ by the scalar product induced by the NNLS compression procedure for (near)exactness degree $2n$ (the resulting weights being positive). Let us term

$$\mathcal{A} = \{Q_k\}_{1 \leq k \leq N} \subset X \;, \quad M > \mathcal{V}(2n,d) = \binom{2n+d}{d} \geq N \tag{27}$$

the set of extracted nodes, and $\boldsymbol{w} = \{w_k\}$ the corresponding weights: the compressed scalar product is

$$\langle f, g \rangle_{\ell^2_{\boldsymbol{w}}(\mathcal{A})} = \sum_{k=1}^{N} w_k\, f(Q_k) g(Q_k) \approx \int_X f g \, d\nu = \sum_{i=1}^{M} f(P_i) g(P_i) = \langle f, g \rangle_{\ell^2(X)} \;, \tag{28}$$

and the corresponding Euclidean norm

$$\|f\|_{\ell^2_{\boldsymbol{w}}(\mathcal{A})} = \sqrt{\langle f, f \rangle_{\ell^2_{\boldsymbol{w}}(\mathcal{A})}} \approx \sqrt{\langle f, f \rangle_{\ell^2(X)}} = \|f\|_{\ell^2(X)} \,. \tag{29}$$

In practice, (28) and (29) are not exact even when f and g are in $\mathbb{P}_n^d$, but the error can be bounded by (13) with $p = fg$ in the fully discrete case $\Omega = X$, $\lambda = \nu$.

Then, we can consider, instead of the orthogonal projection $L_n f$ in $\ell^2(X)$, the orthogonal projection $\mathcal{L}_n f$ in $\ell^2_{\boldsymbol{w}}(\mathcal{A})$, that is

$$\|f - \mathcal{L}_n f\|_{\ell^2_{\boldsymbol{w}}(\mathcal{A})} = \min_{p \in \mathbb{P}_n^d} \|f - p\|_{\ell^2_{\boldsymbol{w}}(\mathcal{A})} \,. \tag{30}$$

Such an orthogonal projection is well-defined and unique, whenever the square Vandermonde-like matrices in $\mathbb{P}_{2n}^d$ corresponding to the extracted node set $\mathcal{A}$ have full rank, since their rectangular restriction to $\mathbb{P}_n^d$ has still full rank. Observe that (30) is a *weighted least-squares* approximation at the discrete set $\mathcal{A}$, cf. [2].

In order to compare $\mathcal{L}_n f$ with $L_n f$, let us now estimate the error made by $\mathcal{L}_n f$ in reconstructing f at the original discrete set X. In this respect, we assume

that the weights are computed working with the discrete orthogonal basis $\boldsymbol{\phi} = \boldsymbol{\psi}$ in the analysis of Section 2, namely in the fully discrete case $\Omega = X$, $\lambda = \nu$.

First, we estimate the $\ell^2(X)$-norm of $\mathcal{L}_n f$. Observe that, setting

$$\mathcal{E} = \int_X (\mathcal{L}_n f)^2 \, d\nu - \sum_{k=1}^{N} w_k \, (\mathcal{L}_n f)^2(Q_k) \, ,$$

we have by (13)

$$\begin{aligned} \|\mathcal{L}_n f\|^2_{\ell^2(X)} = \int_X (\mathcal{L}_n f)^2 \, d\nu = \mathcal{E} + \sum_{k=1}^{N} w_k \, (\mathcal{L}_n f)^2(Q_k) \\ = \mathcal{E} + \|\mathcal{L}_n f\|^2_{\ell^2_{\boldsymbol{w}}(\mathcal{A})} \leq |\mathcal{E}| + \|\mathcal{L}_n f\|^2_{\ell^2_{\boldsymbol{w}}(\mathcal{A})} \\ \leq \|(\mathcal{L}_n f)^2\|_{\ell^2(X)} \, \varepsilon_{mom} + \|\mathcal{L}_n f\|^2_{\ell^2_{\boldsymbol{w}}(\mathcal{A})} \, . \end{aligned}$$

Moreover,

$$\|(\mathcal{L}_n f)^2\|_{\ell^2(X)} \leq \sqrt{M} \, \|\mathcal{L}_n f\|^2_{\ell^\infty(X)} \leq \sqrt{M} \, \|\mathcal{L}_n f\|^2_{\ell^2(X)} \, ,$$

where we have used the fact that we are in a fully discrete situation, so we get

$$\|\mathcal{L}_n f\|_{\ell^2(X)} \leq \frac{\|\mathcal{L}_n f\|_{\ell^2_{\boldsymbol{w}}(\mathcal{A})}}{\sqrt{1 - \sqrt{M} \, \varepsilon_{mom}}} \tag{31}$$

provided that $\sqrt{M} \, \varepsilon_{mom} < 1$. Now, $\mathcal{L}_n$ itself is an orthogonal projection operator on $\mathbb{P}^d_n$, and hence by the Pythagorean theorem

$$\|\mathcal{L}_n f\|_{\ell^2_{\boldsymbol{w}}(\mathcal{A})} \leq \|f\|_{\ell^2_{\boldsymbol{w}}(\mathcal{A})} \leq \sqrt{M} \, \|f\|_{\ell^\infty(\mathcal{A})} \leq \sqrt{M} \, \|f\|_{\ell^\infty(X)} \leq \sqrt{M} \, \|f\|_{L^\infty(K)} \, ,$$

which gives

$$\|\mathcal{L}_n f\|_{\ell^2(X)} \leq \sqrt{\frac{M}{1 - \sqrt{M} \, \varepsilon_{mom}}} \, \|f\|_{L^\infty(K)} \, . \tag{32}$$

Finally, we obtain the estimate

$$\begin{aligned} \|f - \mathcal{L}_n f\|_{\ell^2(X)} \leq \|f - p_n^*\|_{\ell^2(X)} + \|\mathcal{L}_n(f - p_n^*)\|_{\ell^2(X)} \leq \sqrt{M} \, \|f - p_n^*\|_{\ell^\infty(X)} \\ + \sqrt{\frac{M}{1 - \sqrt{M} \, \varepsilon_{mom}}} \, \|f - p_n^*\|_{L^\infty(K)} \leq \sqrt{M} \, \left(1 + \sqrt{\frac{1}{1 - \sqrt{M} \, \varepsilon_{mom}}} \right) \, E_n(f; K) \, , \end{aligned} \tag{33}$$

that is, the reconstruction error in $\ell^2(X)$ by the "compressed" least-squares operator $\mathcal{L}_n$ can be bounded by $E_n(f; K)$ times a constant that has the same order of magnitude of that appearing in (26) for the original "complete" least-squares operator L_n.

We stress two facts. First, we have obtained for compressed least-squares a reconstruction error estimate in the original space $\ell^2(X)$ comparable to that of "complete" least-squares, with a reduction of the sampling points, which can be extremely relevant when $N \ll M$, as in the examples below. Moreover, it is clear that the constraint $\sqrt{M} \, \varepsilon_{mom} < 1$ is quite weak, since it does not require extremely small residuals to keep the approximation quality, differently from the

compression of algebraic cubature formulas of Section 3. These features make compressed polynomial least-squares attractive in applications.

An example of the behavior of compressed least-squares is given in Figure 6 and Table 11. The domain K is the union of three overlapping disks, and the discrete set X is made by the intersection of 5000 Halton points (a low-discrepancy sequence, cf. e.g. [16])) of the smallest rectangle containing the domain, with the domain itself (the resulting cardinality is about 3700).

In Table 11 we display the relative $\ell^2(X)$-errors in the reconstruction of the three test functions (19), by least-squares on the whole X, and by compressed least-squares via the NNLS algorithm, as discussed above. To compute the least-squares approximants, we have solved the overdetermined Chebyshev-Vandermonde system $V\boldsymbol{c} = \boldsymbol{f} = \{f(P_i)\}$ with $V = (\tau_j(P_i))$ (cf. (7)-(8)), and the weighted system $DW\boldsymbol{c}^{compr} = D\boldsymbol{f}^{compr} = D\{f(Q_k)\}$ with $W = (\tau_j(Q_k))$, $D = \mathrm{diag}(\sqrt{w_k})$, by the standard Matlab backslash operator [27]. The relative $\ell^2(X)$-errors on f are $\|f - L_n f\|_{\ell^2(X)}/\|f\|_{\ell^2(X)} = \|\boldsymbol{f} - V\boldsymbol{c}\|_2/\|\boldsymbol{f}\|_2$ for standard least-squares and $\|f - \mathcal{L}_n f\|_{\ell^2(X)}/\|f\|_{\ell^2(X)} = \|\boldsymbol{f} - V\boldsymbol{c}^{compr}\|_2/\|\boldsymbol{f}\|_2$ for compressed least-squares.

It is manifest that the errors have the same order of magnitude (often they are very close to each other, closer than what could be expected from the error estimates). Observe that the moment error ε_{mom} is not as small as in the cubature examples of Section 3. Nevertheless, this does not affect the quality of the compressed approximation, in practice a much higher moment error could be allowed until $\sqrt{M}\varepsilon_{mom} \ll 1$.

Compressed least-squares turn out to be quite efficient: at degree $n = 12$ we have a reasonable error even for the less regular function f_3 with 325 nodes, as if we had sampled on 3700 nodes, that is with only about 9% of the sampling values. We stress that the compression procedure is function independent, so we can select a priori the re-weighted sampling sites on a given region, and then apply the compressed LS formula to different functions. This approach to polynomial least squares could be very useful in applications where the sampling process is difficult or costly.

Remark 2 (LS approximation on Weakly Admissible Meshes). An interesting situation occurs when $X = X_n$ is a *Weakly Admissible Mesh* (WAM) of a compact set K, because in such a case we are able to get an estimate of the uniform norm of the least-squares projection operators. We recall that WAMs are sequences of finite norming sets for a compact set $K \subset \mathbb{R}^d$ (or more generally $K \subset \mathbb{C}^d$), such that a polynomial inequality holds of the form

$$\|p\|_{L^\infty(K)} \le C_n \, \|p\|_{\ell^\infty(X_n)} \, , \quad \forall p \in \mathbb{P}^d_n \, , \tag{34}$$

where both C_n and $\mathrm{card}(X_n)$ increase at most polynomially with n. When C_n is bounded we speak of an Admissible Mesh. Among their properties, we quote that WAMs are preserved by affine transformations, and can be constructed incrementally by finite union and product. Moreover, they are well-suited for uniform least-squares approximation, and for polynomial interpolation at suitable extremal subsets, which are approximate versions of Fekete and Leja points. Concerning these and other basic features of WAMs, we refer the reader to the seminal paper [7], and to [3, 4] among recent developements.

Observe that from the definition of WAM and the Pythagorean theorem we get the chain of inequalities

$$\|L_n f\|_{L^\infty(K)} \le C_n \|L_n f\|_{\ell^\infty(X_n)} \le C_n \|L_n f\|_{\ell^2(X_n)}$$

$$\le C_n \|f\|_{\ell^2(X_n)} \le C_n \sqrt{M}\, \|f\|_{\ell^\infty(X_n)} \le C_n \sqrt{M}\, \|f\|_{L^\infty(K)} \,, \quad \forall f \in C(K) \,,$$

which gives immediately an estimate of the norm of L_n as a projection operator $(C(K), \|\cdot\|_\infty) \to \mathbb{P}_n^d$

$$\|L_n\| \le C_n \sqrt{M} \,. \tag{35}$$

On the other hand, concerning the compressed least-squares operator $\mathcal{L}_n$, from the definition of WAM and (32) we have

$$\|\mathcal{L}_n f\|_{L^\infty(K)} \le C_n \|\mathcal{L}_n f\|_{\ell^\infty(X_n)} \le C_n \|\mathcal{L}_n f\|_{\ell^2(X_n)}$$

$$\le C_n \sqrt{\frac{M}{1 - \sqrt{M}\, \varepsilon_{mom}}}\, \|f\|_{L^\infty(K)} \,, \quad \forall f \in C(K) \,,$$

from which we obtain the operator norm bound

$$\|\mathcal{L}_n\| \le C_n \sqrt{\frac{M}{1 - \sqrt{M}\, \varepsilon_{mom}}} \,, \tag{36}$$

which is close to (35) when $\varepsilon_{mom} \ll 1/\sqrt{M}$. A standard computation for projection operators $\mathbb{P}_n : (C(K), \|\cdot\|_\infty) \to \mathbb{P}_n^d$ shows that

$$\|f - \mathcal{P}_n f\|_{L^\infty(K)} \le (1 + \|\mathcal{P}_n\|)\, E_n(f; K) \,, \tag{37}$$

where either $\mathcal{P}_n = L_n$ and (35) applies, or $\mathcal{P}_n = \mathcal{L}_n$ and (36) applies.

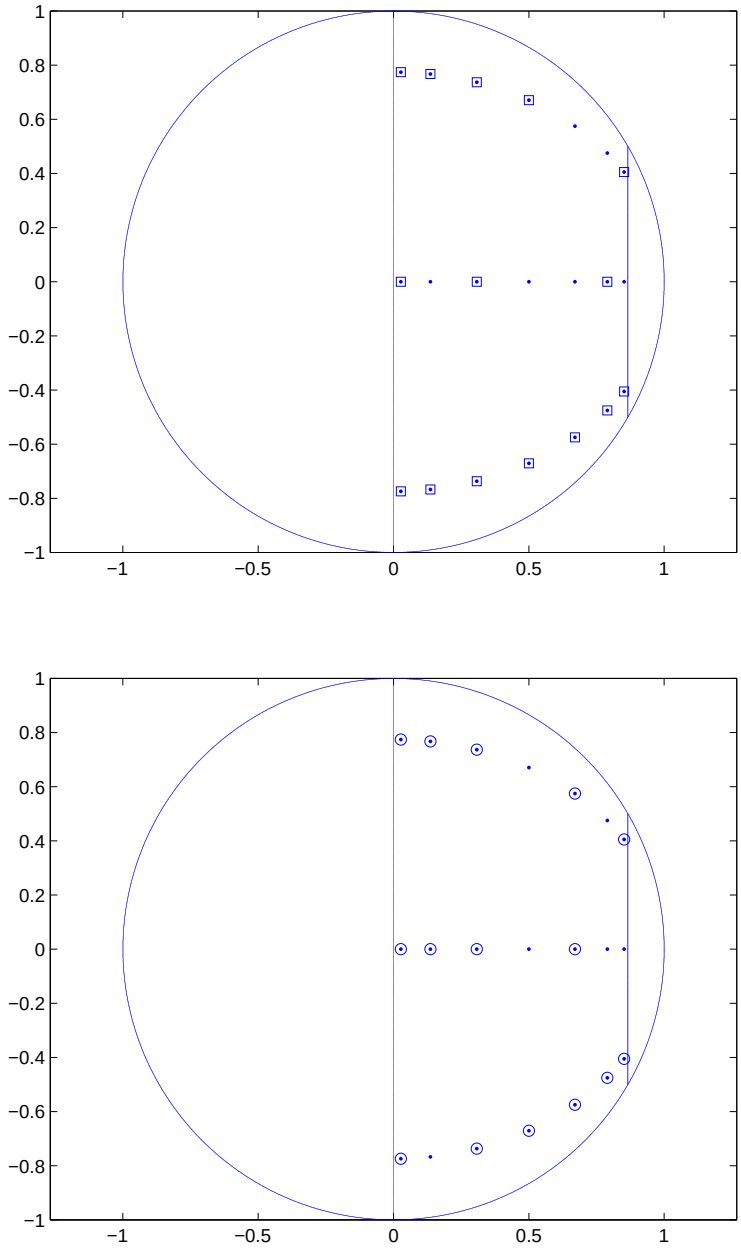

Figure 1: $21 = 3 \times 7$ product quadrature nodes $(\cdot)$ of algebraic exactness degree 4 on a circular zone of the unit disk with, and compression into 15 points, by QR with column pivoting (top, $(\square)$) and NNLS (bottom, $(\circ)$).

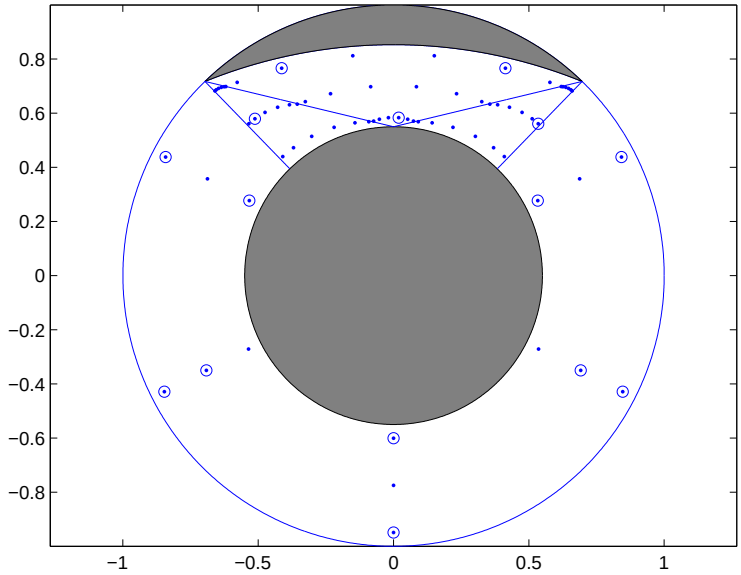

Figure 2: 69 cubature nodes (·) of algebraic exactness degree 4 on an obscured and vignetted pupil, and compression into 15 points (○) by NNLS.

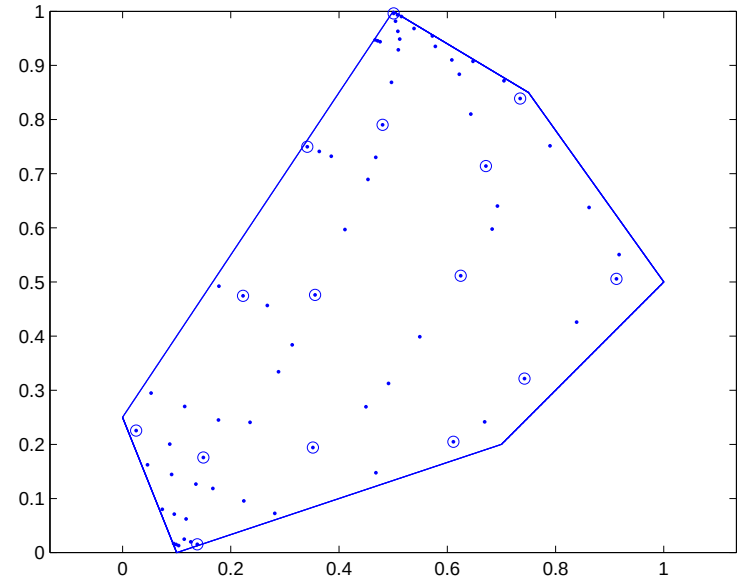

Figure 3: 72 cubature nodes (·) of algebraic exactness degree 4 on a convex hexagon, and compression into 15 points (○) by NNLS.

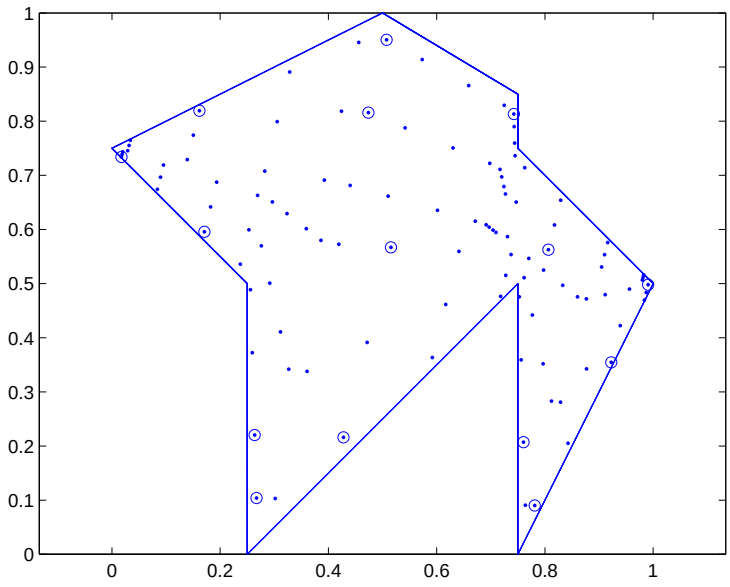

Figure 4: 108 cubature nodes (·) of algebraic exactness degree 4 on a nonconvex nonagon, and compression into 15 points (○) by QR with column pivoting.

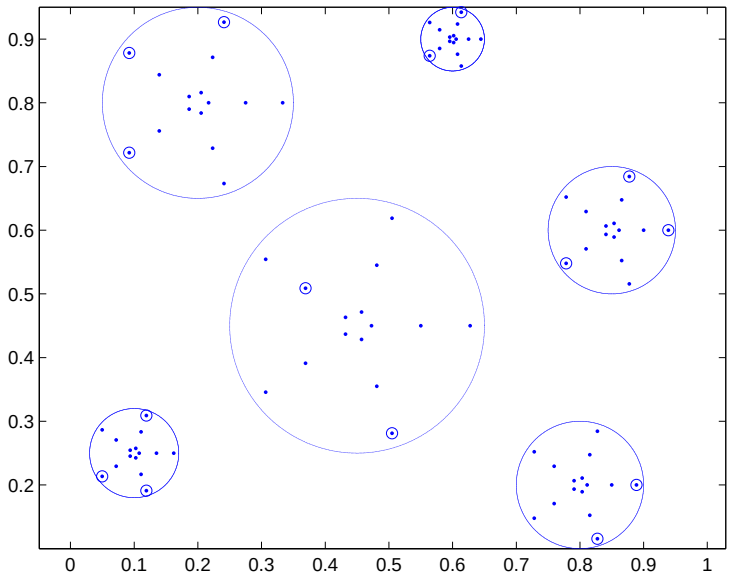

Figure 5: 90 cubature nodes (·) of algebraic exactness degree 4 on the union of six disconnected disks, and compression into 15 points (○) by QR with column pivoting.

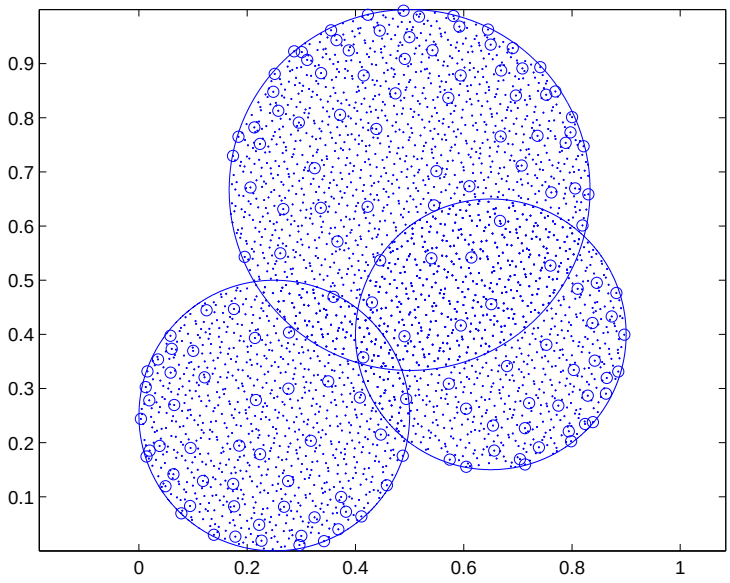

Figure 6: 153 nodes (∘) for compressed least-squares of degree 8, extracted by NNLS from about 3700 Halton points (·) of the union of three disks.

References

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

[2] A. Björk, Numerical methods for least squares problems, SIAM, 1996.

[3] L. Bos, J.P. Calvi, N. Levenberg, A. Sommariva and M. Vianello, Geometric Weakly Admissible Meshes, Discrete Least Squares Approximations and Approximate Fekete Points preprint, Math. Comp. 80 (2011), 1601–1621.

[4] L. Bos, S. De Marchi, A. Sommariva and M. Vianello, Computing multivariate Fekete and Leja points by numerical linear algebra, SIAM J. Numer. Anal. 48 (2010), 1984–1999.

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

[6] P.A. Businger and G.H. Golub, Linear least-squares solutions by Householder transformations, Numer. Math. 7 (1965), 269–276.

[7] J.P. Calvi and N. Levenberg, Uniform approximation by discrete least squares polynomials, J. Approx. Theory 152 (2008), 82–100.

[8] A. Civril and M. Magdon-Ismail, On Selecting a Maximum Volume Submatrix of a Matrix and Related Problems, Theoretical Computer Science 410 (2009), 4801–4811.

Table 1: Compression of the algebraic cubature formula on the circular zone of Figure 1; M is the cardinality of the original formula, N that of the compressed formula, ε_{mom} the moment reconstruction error in (6).

	deg n	5	10	15	20	25	30
	old card M	24	78	144	253	364	528
	new card N	21	66	136	231	351	496
QRpiv	$\sum \lvert w_k \rvert / \lvert \sum w_k \rvert$	1.02	1.08	1.11	1.41	1.08	1.06
	ε_{mom}	1e-15	1e-15	2e-15	2e-15	2e-15	3e-15
	cputime (sec)	0.01	0.02	0.07	0.10	0.14	0.79
NNLS	ε_{mom}	1e-15	1e-15	2e-15	3e-15	6e-15	8e-15
	cputime (sec)	0.16	0.04	0.26	1.59	7.83	32.30

Table 2: Relative error in the integration of the three test functions on the circular zone of Figure 1 by the compressed formulas.

	deg n	5	10	15	20	25	30
f_1	α	2.3e02	9.3e01	7.0e02	1.6e03	6.3e03	1.7e04
	QRpiv	1e-15	1e-15	2e-15	3e-15	1e-14	3e-15
	NNLS	2e-15	2e-15	7e-16	9e-15	9e-15	5e-15
f_2	α	2.2e00	2.1e00	2.1e00	2.1e00	2.1e00	2.1e00
	QRpiv	3e-04	1e-07	2e-10	5e-15	1e-16	1e-15
	NNLS	3e-04	1e-07	2e-10	5e-15	0	2e-15
f_3	α	3.7e00	5.0e00	5.3e00	5.5e00	5.6e00	5.6e00
	QRpiv	3e-03	4e-05	1e-05	9e-08	1e-06	3e-07
	NNLS	3e-03	6e-05	1e-05	2e-06	1e-06	2e-07

Table 3: As in Table 1 for the obscured and vignetted pupil of Figure 2.

	deg n	5	10	15	20	25	30
	old card M	108	282	603	957	1498	2032
	new card N	21	66	136	231	351	496
QRpiv	$\sum \lvert w_k \rvert / \lvert \sum w_k \rvert$	1.64	1.15	1.28	1.51	1.14	1.22
	ε_{mom}	3e-15	5e-15	5e-15	7e-15	8e-15	1e-14
	cputime (sec)	0.01	0.07	0.22	0.53	0.88	2.03
NNLS	ε_{mom}	1e-15	3e-15	5e-15	1e-14	9e-15	1e-14
	cputime (sec)	0.02	0.10	0.43	2.35	9.17	42.20

Table 4: Relative error in the integration of the three test functions on the obscured and vignetted pupil of Figure 2 by the compressed formulas.

	deg n	5	10	15	20	25	30
f_1	α	8.0e01	1.1e02	4.8e03	6.2e03	2.2e05	2.9e05
	QRpiv	9e-15	5e-15	4e-14	3e-14	3e-12	3e-12
	NNLS	9e-15	6e-15	1e-13	1e-13	6e-13	3e-12
f_2	α	1.9e00	1.8e00	1.8e00	1.8e00	1.8e00	1.8e00
	QRpiv	2e-03	3e-07	4e-12	5e-15	6e-15	7e-15
	NNLS	1e-03	2e-07	9e-12	8e-15	7e-15	7e-15
f_3	α	2.6e00	3.0e00	3.0e00	3.1e00	3.1e00	3.1e00
	QRpiv	6e-04	5e-05	3e-07	7e-07	8e-08	1e-07
	NNLS	6e-04	4e-06	1e-06	4e-08	2e-08	5e-08

Table 5: As in Table 1 for the convex hexagon of Figure 3.

	deg n	5	10	15	20	25	30
	old card M	72	252	432	792	1092	1632
	new card N	21	66	136	231	351	496
QRpiv	$\sum \lvert w_k \rvert / \lvert \sum w_k \rvert$	1.20	1.09	1.05	1.16	1.20	1.25
	ε_{mom}	4e-16	7e-16	8e-16	1e-15	1e-15	1e-15
	cputime (sec)	0.01	0.08	0.13	0.55	1.45	3.57
NNLS	ε_{mom}	2e-16	7e-16	1e-15	2e-15	3e-15	5e-15
	cputime (sec)	0.02	0.06	0.43	2.47	10.83	45.71

Table 6: Relative error in the integration of the three test functions on the convex hexagon of Figure 3 by the compressed formulas.

	deg n	5	10	15	20	25	30
f_1	α	1.8e01	8.8e01	3.7e02	1.5e03	5.8e03	2.1e04
	QRpiv	2e-15	8e-15	1e-14	6e-15	1e-14	1e-13
	NNLS	3e-15	8e-15	2e-14	5e-15	3e-14	5e-15
f_2	α	5.3e00	5.2e00	5.2e00	5.1e00	5.1e00	5.1e00
	QRpiv	2e-05	4e-11	3e-15	1e-15	3e-15	3e-15
	NNLS	3e-05	1e-10	2e-15	2e-15	3e-15	2e-15
f_3	α	2.1e01	2.4e01	2.4e01	2.5e01	2.5e01	2.5e01
	QRpiv	4e-03	1e-04	4e-06	2e-06	3e-06	8e-07
	NNLS	6e-03	4e-04	2e-05	3e-06	3e-07	1e-06

Table 7: As in Table 1 for the nonconvex nonagon of Figure 4.

	deg n	5	10	15	20	25	30
	old card M	108	378	648	1188	1638	2448
	new card N	21	66	136	231	351	496
QRpiv	$\sum \lvert w_k \rvert / \lvert \sum w_k \rvert$	1.20	1.09	1.05	1.16	1.20	1.25
	ε_{mom}	4e-16	7e-16	8e-16	1e-15	1e-15	1e-15
	cputime (sec)	0.01	0.05	0.12	2.54	1.47	3.59
NNLS	ε_{mom}	2e-16	5e-16	2e-15	3e-15	5e-15	6e-15
	cputime (sec)	0.01	0.11	0.50	2.85	11.43	47.97

Table 8: Relative error in the integration of the three test functions on the nonconvex nonagon of Figure 4 by the compressed formulas.

	deg n	5	10	15	20	25	30
f_1	α	2.0e01	1.0e02	4.7e02	2.0e03	8.4e03	3.3e04
	QRpiv	2e-15	3e-15	3e-15	3e-15	3e-14	2e-13
	NNLS	2e-15	2e-15	5e-15	1e-14	3e-14	2e-13
f_2	α	5.9e00	5.7e00	5.7e00	5.7e00	5.7e00	5.7e00
	QRpiv	1e-05	3e-11	0	1e-15	2e-15	2e-15
	NNLS	4e-06	3e-11	1e-16	1e-15	2e-15	2e-15
f_3	α	2.3e01	2.8e01	2.9e01	2.9e01	2.9e01	3.0e01
	QRpiv	2e-03	1e-04	1e-05	4e-06	1e-07	2e-06
	NNLS	2e-03	9e-05	8e-05	1e-05	5e-07	4e-07

Table 9: As in Table 1 for the disconnected disk union of Figure 5.

	deg n	5	10	15	20	25	30
	old card M	144	396	864	1386	2184	2976
	new card N	21	66	136	231	351	496
QRpiv	$\sum \lvert w_k \rvert / \lvert \sum w_k \rvert$	1.05	1.13	1.20	1.09	1.22	1.15
	ε_{mom}	1e-16	2e-16	4e-16	8e-16	4e-16	2e-15
	cputime (sec)	0.01	0.10	0.14	0.46	2.65	3.17
NNLS	ε_{mom}	9e-17	2e-16	5e-16	9e-16	2e-08	2e-15
	cputime (sec)	0.02	0.17	0.46	2.35	12.42	42.35

Table 10: Relative error in the integration of the three test functions on the disconnected disk union of Figure 5 by the compressed formulas.

	deg n	5	10	15	20	25	30
f_1	α	1.1e02	3.4e02	9.6e02	2.7e03	7.6e03	2.1e04
	QRpiv	2e-15	9e-16	4e-15	6e-15	9e-15	8e-15
	NNLS	6e-16	2e-16	5e-15	4e-15	4e-08	6e-15
f_2	α	7.0e01	7.4e01	7.4e01	7.5e01	7.5e01	7.5e01
	QRpiv	2e-05	2e-12	2e-15	4e-15	2e-14	1e-14
	NNLS	2e-05	4e-11	3e-15	5e-15	9e-09	1e-14
f_3	α	3.6e01	3.5e01	3.5e01	3.5e01	3.5e01	3.5e01
	QRpiv	4e-03	9e-05	8e-06	7e-06	2e-06	1e-06
	NNLS	3e-03	1e-04	7e-06	2e-06	5e-06	7e-07

Table 11: Relative errors in $\ell^2(X)$ for least-squares approximation of the three test functions on the disk union of Figure 6; here $M = \mathrm{card}(X) \approx 3700$.

	deg n	2	4	6	8	10	12
	N	15	45	91	153	231	325
	N/M	0.004	0.012	0.015	0.041	0.062	0.088
	ε_{mom}	4.9e-14	1.2e-13	3.4e-13	4.3e-13	8.8e-13	2.5e-12
f_1	LS	1.2e-15	1.8e-15	1.3e-15	1.5e-15	3.7e-15	4.7e-15
	compr	3.1e-16	6.2e-16	8.3e-15	1.2e-15	1.6e-15	2.7e-15
f_2	LS	3.0e-03	8.2e-05	1.7e-06	2.7e-08	3.5e-10	3.9e-12
	compr	3.3e-03	8.6e-05	1.7e-06	2.7e-08	3.5e-10	3.9e-12
f_3	LS	1.2e-01	1.5e-02	4.2e-03	1.6e-03	7.2e-04	3.7e-04
	compr	1.3e-01	1.6e-02	4.4e-03	1.7e-03	7.6e-04	3.8e-04

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

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

[11] G. Da Fies, A. Sommariva and M. Vianello, SUBP: Matlab package for subperiodic trigonometric quadrature and multivariate applications, online at: `http://www.math.unipd.it/~marcov/CAAsoft.html`.

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

[13] G. Da Fies and M. Vianello, Algebraic cubature on planar lenses and bubbles, Dolomites Res. Notes Approx. 5 (2012), 7–12.

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

[15] P.J. Davis, A construction of nonnegative approximate quadratures, Math. Comp. 21 (1967), 578–582.

[16] M. Drmota and R.F. Tichy, Sequences, discrepancies and applications, Lecture Notes in Mathematics, 1651, Springer, Berlin, 1997.

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

[18] M. Festa and A. Sommariva, Computing almost minimal formulas on the square, J. Comput. Appl. Math. 236 (2012), 4296–4302.

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

[20] W. Gautschi, Orthogonal polynomials (in Matlab), J. Comput. Appl. Math. 178 (2005), 215–234, software online at:
`http://www.cs.purdue.edu/archives/2002/wxg/codes`.

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

[22] 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.

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

[24] D. Huybrechs, Stable high-order quadrature rules with equidistant points, J. Comput. Appl. Math. 231 (2009), 933–947.

[25] C.L. Lawson and R.J. Hanson, Solving least squares problems, Classics in Applied Mathematics 15, SIAM, Philadelphia, 1995.

[26] C.-J. Li, P. Lamberti and C. Dagnino, Numerical integration over polygons using an eight-node quadrilateral spline finite element, J. Comput. Appl. Math. 233 (2009), 279–292.

[27] The Mathworks, MATLAB documentation set, 2014 version (available online at: `http://www.mathworks.com`).

[28] S.E. Mousavi and H. Xiao and N. Sukumar, Generalized Gaussian quadrature rules on arbitrary polygons, Internat. J. Numer. Methods in Engrg. 82 (2010), 99–113.

[29] I.P. Omelyan and V.B. Solovyan, Improved cubature formulae of high degrees of exactness for the square, J. Comput. Appl. Math. 188 (2006), 190–204.

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

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

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

[33] A. Sommariva and M. Vianello, Computing approximate Fekete points by QR factorizations of Vandermonde matrices, Comput. Math. Appl. 57 (2009), 1324–1336.

[34] N. Sundararajan, S. Bordas and D. Roy Mahapatra, Numerical integration over arbitrary polygonal domains based on Schwarz-Christoffel conformal mapping, Internat. J. Numer. Methods Engrg. 80 (2009), 103–134.

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

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

[37] V. Tchakaloff, Formules de cubature mécaniques à coefficients nonnégatifs, Bull. Sci. Math. 81 (1957), 123–134.

[38] 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.

Appendix: Matlab functions for the compression of a bivariate discrete measure

```
function [pts,w,momerr] = compresscub(deg,X,omega,pos)
% INPUT:
% deg: polynomial exactness degree
% X: 2-column array of point coordinates
% omega: 1-column array of weights
% pos: NNLS for pos=1, QR with column pivoting for pos=0
% OUTPUT:
% pts: 2-column array of extracted points
% w: 1-column array of corresponding weights (positive for pos=1)
% momerr: moment reconstruction error

% total-degree Chebyshev-Vandermonde matrix at X
rect=[min(X(:,1)) max(X(:,1)) min(X(:,2)) max(X(:,2))];
V=chebvand(deg,X,rect);
% polynomial basis orthogonalization
[Q,R]=qr(V,0);
% tiny imaginary parts may appear
Q=real(Q);
% possible re-orthogonalization
% [Q,R1]=qr(Q,0);
% moments of the orthogonal basis
orthmom=Q'*omega;
% weigths computation
if pos == 1
weights=lsqnonneg(Q',orthmom);
else
weights=Q'\orthmom;
end;
% indexes of nonvanishing weights and compression
ind=find(abs(weights)>0);
pts=X(ind,:);
w=weights(ind);
% moment reconstruction error
% bivariate Chebyshev basis
mom=V'*omega;
momerr=norm(V(ind,:)'*w-mom);
% discrete OP basis
% momerr=norm(Q(ind,:)'*w-orthmom);
end


function V = chebvand(deg,X,rect);
% INPUT:
% deg = polynomial degree
% X = 2-column array of point coordinates
% rect = 4-component vector such that the rectangle
% [rect(1),rect(2)] x [rect(3),rect(4)] contains X
% OUTPUT:
% V = Chebyshev-Vandermonde matrix at X

% couples with length less or equal to deg
j=(0:1:deg);
[j1,j2]=meshgrid(j);
good=find(j1(:)+j2(:)<deg+1);
couples=[j1(good) j2(good)];
% mapping X in the square [-1,1]^2
a=rect(1);b=rect(2);c=rect(3);d=rect(4);
map=[(2*X(:,1)-b-a)/(b-a) (2*X(:,2)-d-c)/(d-c)];
% total-degree Chebyshev-Vandermonde matrix at X
V1=cos(couples(:,1)*acos(map(:,1)'));
V2=cos(couples(:,2)*acos(map(:,2)'));
V=(V1.*V2)';
end
```