Polynomial interpolation and cubature over polygons *

M. Gentile, A. Sommariva and M. Vianello[1]
Dept. of Pure and Applied Mathematics
University of Padova (Italy)

April 23, 2011

Abstract

We have implemented a Matlab code to compute Discrete Extremal Sets (of Fekete and Leja type) on convex or concave polygons, together with the corresponding interpolatory cubature formulas. The method works by QR and LU factorizations of rectangular Vandermonde matrices on Weakly Admissible Meshes (WAMs) of polygons, constructed by polygon quadrangulation.

2000 AMS subject classification: 65D05, 65D32.

Keywords: Weakly Admissible Meshes, Discrete Extremal Sets, Approximate Fekete Points, Discrete Leja Points, Polynomial Interpolation, Algebraic Cubature, Quadrangles, Polygon Quadrangulation.

1 Introduction.

Global polynomial approximation is still a challenging topic in the multivariate setting. The geometry of the interpolation domain and of its discrete models play a key role, substantially not yet well understood.

In particular, few results are known about the so-called Fekete points of a d-dimensional compact domain, i.e., points that maximize the Vandermonde determinant for a given interpolation degree n, ensuring an (at most) algebraic growth of the Lebesgue constant in the corresponding interpolation process; cf. e.g. [2, 5]. Moreover, the numerical computation of Fekete points becomes rapidly a very large scale nonlinear optimization problem in $N \times d$ variables, where $N \sim n^d/d!$ is the dimension of the polynomial space (cf., e.g., [25, 29]).

A reasonable approach for the computation of Fekete points is to use a discretization of the domain, moving from continuous to combinatorial optimization. Good discrete models of general compact sets are provided by the so-called "Weakly Admissible Meshes", recently studied by Calvi and Levenberg in [11].

*Supported by the "ex-60%" funds and by the project "Interpolation and Extrapolation: new algorithms and applications" of the University of Padova, and by the INdAM GNCS.

[1]Corresponding author, e-mail: marcov@math.unipd.it

Given a polynomial determining compact set $K \subset \mathbb{R}^d$ or $K \subset \mathbb{C}^d$ (i.e., polynomials vanishing there are identically zero), a *Weakly Admissible Mesh* (WAM) is defined in [11] to be a sequence of discrete subsets $\mathcal{A}_n \subset K$ such that

$$\|p\|_K \leq C(\mathcal{A}_n)\|p\|_{\mathcal{A}_n}\ ,\ \ \forall p \in \mathbb{P}_n^d \tag{1}$$

where both $\mathrm{card}(\mathcal{A}_n) \geq N$ and $C(\mathcal{A}_n)$ grow at most *polynomially* with n (we use the notation $\|f\|_X = \sup_{x \in X} |f(x)|$ for f bounded function on the compact X). When $C(\mathcal{A}_n)$ is bounded we speak of an Admissible Mesh (AM).

Among their properties, it is worth recalling the following ones (cf. [11]), which can be considered as a recipe to construct new from known WAMs:

(I) if α is a *polynomial vector mapping* of degree m and $\mathcal{A}_n$ a WAM for K, then $\alpha(\mathcal{A}_{nm})$ is a WAM on $\alpha(K)$ with constant $C(\mathcal{A}_{nm})$;

(II) any sequence of unisolvent *interpolation sets* whose Lebesgue constant Λ_n grows at most polynomially with n is a WAM, with constant $C(\mathcal{A}_n) = \Lambda_n$;

(III) a *finite product* of WAMs is a WAM (even for tensor-product polynomials) on the corresponding product of compacts, $C(\mathcal{A}_n)$ being the product of the corresponding constants;

(IV) a *finite union* of WAMs is a WAM on the corresponding union of compacts, $C(\mathcal{A}_n)$ being the maximum of the corresponding constants.

The usefulness of WAMs in polynomial approximation is given by the following properties (cf. [3, 11]):

(V) the *least-squares* polynomial $\mathcal{L}_{\mathcal{A}_n} f$ on a WAM, $f \in C(K)$, is such that

$$\|f - \mathcal{L}_{\mathcal{A}_n} f\|_K \lessapprox C(\mathcal{A}_n)\sqrt{\mathrm{card}(\mathcal{A}_n)}\ \min\{\|f-p\|_K,\, p \in \mathbb{P}_n^d\}\ .$$

(VI) *Fekete points extracted* from WAMs have a Lebesgue constant bounded by $NC(\mathcal{A}_n)$, and are asymptotically distributed as the continuum Fekete points.

A key feature, which is appealing for computational purposes, is the availability of low cardinality WAMs for many standard compact sets in dimension $d = 2, 3$; cf. [3, 6, 14]. These WAMs typically have $\mathcal{O}(n^d)$ points, the same order of growth of the cardinality $N = N_n = \dim(\mathbb{P}_n^d) = \binom{n+d}{d}$ of any unisolvent interpolation set (being $\mathbb{P}_n^d$-determining, the cardinality of a WAM is necessarily not smaller than the dimension N of the polynomial space).

In the next Section we recall two recent algorithms that compute Fekete-like sets for (multivariate) polynomial interpolation, called *Discrete Extremal Sets*, starting from a WAM of a compact set. Then, in Section 3 we show how to construct efficiently a WAM of a convex quadrangle. In Section 4 we describe and test the method underlying our Matlab code (POLYGINT), that computes Discrete Extremal Sets of simple polygons via (coarse) convex quadrangulation, together with the corresponding interpolatory cubature formulas. Using a quadrangulation instead of a more usual triangulation, as was done in [4] for interpolation, allows to work with WAMs that have on the average half the number of mesh points. We stress that the cardinality of the extracted Discrete Extremal Sets is $N = (n+1)(n+2)/2$ irrespectively of the shape of the polygon and of the number of polygon sides. The code is available at [18].

2 Fekete Points and Discrete Extremal Sets.

We briefly recall the concept of Fekete points for polynomial interpolation. It is worth observing that such Fekete points should not be confused with the "minimum energy" Fekete points, the two concepts being equivalent only in the univariate complex case (cf. [24]).

Given a compact set $K \subset \mathbb{R}^d$ (or $\mathbb{C}^d$), a basis $\boldsymbol{p} = \{p_1, \dots, p_N\}$ of $\mathbb{P}_n^d$ and a finite set $\boldsymbol{\xi} = \{\xi_1, \dots, \xi_N\} \subset K$, ordering in some manner the points and the basis we can construct the Vandermonde-like matrix $V(\boldsymbol{\xi}, \boldsymbol{p}) = [p_j(\xi_i)]$, $1 \leq i, j \leq N$. If $\det V(\boldsymbol{\xi}, \boldsymbol{p}) \neq 0$ the set $\{\xi_1, \dots, \xi_N\}$ is unisolvent for interpolation, and the polynomials

$$\ell_j(x) = \frac{\det V(\xi_1, \dots, \xi_{j-1}, x, \xi_{j+1}, \dots, \xi_N, \boldsymbol{p})}{\det V(\xi_1, \dots, \xi_{j-1}, \xi_j, \xi_{j+1}, \dots, \xi_N, \boldsymbol{p})}\ , \quad j = 1, \dots, N\ , \tag{2}$$

form a *cardinal basis*, i.e., $\ell_j(\xi_k) = \delta_{jk}$ and

$$L_n f(x) = \sum_{j=1}^{N} f(\xi_j)\, \ell_j(x) \tag{3}$$

is the polynomial in $\mathbb{P}_n^d$ that interpolates a function f at $\{\xi_1, \dots, \xi_N\}$.

In the case that such points maximize the (absolute value of the) denominator of (2) in K^N (Fekete points), then $\|\ell_j\|_\infty \leq 1$ for every j, and thus the Lebesgue constant (the norm of the interpolation operator) is bounded by the dimension of the interpolation space,

$$\Lambda_n = \|L_n\| = \max_{x \in K} \sum_{j=1}^{N} |\ell_j(x)| \leq N\ . \tag{4}$$

Clearly, Fekete points as well as Lebesgue constants are independent of the choice of the basis, since the determinant of the Vandermonde-like matrices changes by a factor independent of the points (namely the determinant of the transformation matrix between the bases). Moreover, Fekete points and Lebesgue constants are preserved under affine mapping of the domain. It is also worth recalling that (4) is often a rather pessimistic overestimate of the actual growth.

There are several open problems about Fekete points, whose properties have been studied till now mainly in the univariate complex case in view of their deep connection with potential theory. They are analytically known only in few cases: the interval (Gauss-Lobatto points), the complex circle (equispaced points), and the cube (tensor-product of Gauss-Lobatto points for tensor interpolation). An important qualitative result has been proved only recently, namely that Fekete points are asymptotically equidistributed with respect the pluripotential equilibrium measure of K, cf. [2]. Their asymptotic spacing is known only in few instances, cf. the recent paper [5]. Moreover, the numerical computation of Fekete points becomes rapidly a very large scale problem, namely a nonlinear optimization problem in $N \times d$ variables. It has been solved numerically only in very special cases, like the triangle and the sphere, for a fixed limited range of degrees.

A reasonable approach for the computation of Fekete points is to use a discretization of the domain, moving from the continuum to nonlinear combinatorial optimization. Property **(VI)** gives a first guideline on the fact that WAMs are good candidates as starting meshes. Consider the rectangular Vandermonde-like matrix associated to a WAM

$$V(\boldsymbol{a},\boldsymbol{p}) = V(a_1,\dots,a_M;p_1,\dots,p_N) = [p_j(a_i)]\ ,\ \ 1 \le i \le M\ ,\ 1 \le j \le N \quad (5)$$

where $\boldsymbol{a} = (a_i)$ is the array of mesh points, and $\boldsymbol{p} = (p_j)$ is the array of basis polynomials for $\mathbb{P}_n^d$. For convenience, we shall consider $\boldsymbol{p}$ as a column vector $\boldsymbol{p} = (p_1,\dots,p_N)^t$. Since the rows of the matrix $V(\boldsymbol{a},\boldsymbol{p})$ correspond to the mesh points and the columns to the basis elements, computing the Fekete points of a WAM amounts to selecting N rows of V such that the volume generated by these rows, i.e., the absolute value of the determinant of the resulting $N \times N$ submatrix, is maximum.

This problem, however, is known to be NP-hard, so heuristic or stochastic algorithms are mandatory; cf. [13] for the notion of volume generated by a set of vectors (which generalizes the geometric concept related to parallelograms and parallelepipeds), and an analysis of the problem from a computational complexity point of view.

Almost surprisingly, good approximate solutions, called *Discrete Extremal Sets*, can be given by basic procedures of numerical linear algebra. The first, which gives the *Approximate Fekete Points*, corresponds to a greedy maximization of submatrix volumes:

greedy volume maximization (AFP: Approximate Fekete Points)
- $V = V(\boldsymbol{a},\boldsymbol{p})$; $ind = [\,]$;
- `for` $k = 1:N$ "select i_k: $\mathrm{vol}\,V([ind,i_k]\,,1:N)$ is max"; $ind = [ind,i_k]$; `end`
- $\boldsymbol{\xi} = \boldsymbol{a}(i_1,\dots,i_N)$

Its core "select i_k: $\mathrm{vol}\,V([ind,i_k]\,,1:N)$ is maximum" can be implemented as "select the largest norm row $row_{i_k}(V)$ and remove from every row of V its orthogonal projection onto row_{i_k}". This is equivalent to the *QR factorization with column pivoting* (cf. [10]) of the transpose Vandermonde matrix, since such process automatically seeks the maximum keeping invariant the volumes by column orthogonalization (a simple geometric interpretation of the method is that of a greedy maximization of the parallelepiped volume generated by three in a bunch of 3d vectors). This factorization is just that applied in Matlab for the solution of underdetermined systems by the "backslash" operator, as in the following Matlab-like script:

AFP code
- $W = V^t(\boldsymbol{a},\boldsymbol{p})$; $\boldsymbol{b} = \texttt{ones}(1:N)$; $\boldsymbol{w} = W\backslash\boldsymbol{b}$; $ind = \texttt{find}(\boldsymbol{w} \neq \boldsymbol{0})$; $\boldsymbol{\xi} = \boldsymbol{a}(ind)$

The second algorithm, which gives the *Discrete Leja Points*, corresponds to a greedy maximization of nested square submatrix determinants:

greedy determinant maximization (DLP: Discrete Leja Points)
- $V = V(\boldsymbol{a},\boldsymbol{p})$; $ind = [\,]$;
- `for` $k = 1:N$ "select i_k: $|\det V([ind,i_k]\,,1:k)|$ is max"; $ind = [ind,i_k]$; `end`
- $\boldsymbol{\xi} = \boldsymbol{a}(i_1,\dots,i_N)$

Its core, "select i_k: $|\det V([ind, i_k], 1:k)|$ is maximum", can be implemented by one column elimination step of the Gaussian elimination process with standard row pivoting, since such process automatically seeks the maximum keeping invariant the absolute value of the relevant subdeterminants. Here is the corresponding Matlab-like script based on *LU factorization with row pivoting*:

DLP code
- $V = V(\boldsymbol{a}, \boldsymbol{p})$; $[L, U, \boldsymbol{\sigma}] = \texttt{LU}(V, \text{"}\texttt{vector}\text{"})$; $ind = \boldsymbol{\sigma}(1:N)$; $\boldsymbol{\xi} = \boldsymbol{a}(ind)$

Notice that in algorithm AFP we could take as right-hand side $\boldsymbol{b}$ any nonzero vector, and that in algorithm DLP we are using the version of the LU factorization with row pivoting that produces a row permutation vector $\boldsymbol{\sigma}$. See [3, 4, 27] and the references therein for a complete discussion of these two approaches.

When the conditioning of the Vandermonde matrices is too high, the algorithms can still be used provided that a preliminary iterated orthogonalization is performed, as in the following:

iterated orthogonalization code
- $V = V(\boldsymbol{a}, \boldsymbol{q})$; $[Q_1, R_1] = \texttt{qr}(V, 0)$; $[Q_2, R_2] = \texttt{qr}(Q_1, 0)$; $T = \texttt{inv}(R_2 * R_1)$

which amounts to a change of polynomial basis from $\boldsymbol{q}$ to $\boldsymbol{p} = T^t\boldsymbol{q}$ which is orthonormal with respect to the discrete inner product $\langle f, g\rangle = \sum_{i=1}^{M} f(a_i)\overline{g(a_i)}$ (we use here the QR factorization with Q rectangular $M \times N$ and R upper triangular $N \times N$). Observe that the Vandermonde matrix in the new basis,

$$V(\boldsymbol{a}, \boldsymbol{p}) = V(\boldsymbol{a}, \boldsymbol{q})T = Q_2 \tag{6}$$

is a numerically orthogonal (unitary) matrix, i.e., $\overline{Q_2^t}Q_2 = I$. Two orthogonalization iterations generally suffice, unless the original matrix $V(\boldsymbol{a}, \boldsymbol{q})$ is so severely ill-conditioned (condition number greater than the reciprocal of machine precision) that the algorithm fails. This phenomenon is well-known in numerical Gram-Schmidt orthogonalization, cf. [19].

Once the underlying extraction WAM has been fixed, differently from the continuum Fekete points, Approximate Fekete Points depend on the choice of the basis, and Discrete Leja Points depend also on its order. An important feature is that Discrete Leja Points form a *sequence*, i.e., if the basis $\boldsymbol{p}$ is such that its first $N_k = \dim(\mathbb{P}_k^d)$ elements span $\mathbb{P}_k^d$, $1 \le k \le n$, then the first N_k Discrete Leja Points are a unisolvent set for interpolation in $\mathbb{P}_k^d$.

Under the latter assumption for Discrete Leja Points, the two families of Discrete Extremal Sets share the same asymptotic behavior, which is exactly that of the continuum Fekete points: the corresponding discrete measures converge weak-$*$ to the pluripotential equilibrium measure of the underlying compact set, cf. [2, 3, 4].

3 Quadrangle WAMs.

It is well-known that any *convex* quadrangle with vertices P_1, P_2, P_3, P_4 is the image of a *bilinear transformation* of the square, namely

$$(x, y) = \sigma(u, v) = \frac{1}{4}\,((1-u)(1-v)P_1 + (1+u)(1-v)P_2$$

$$+(1+u)(1+v)P_3 + (1-u)(1+v)P_4) \ , \quad (u,v) \in [-1,1]^2 \qquad (7)$$

with a triangle, e.g. $P_3 = P_4$, as a special degenerate case. We can then prove the following

Proposition 1 *The sequence of "oblique" Chebyshev-Lobatto grids*

$$\mathcal{A}_n = \{\sigma(\xi_j, \xi_k) \, , \, 0 \le j,k \le n\} \ , \quad \xi_s = \cos \frac{s\pi}{n} \qquad (8)$$

is a WAM (cf. (1)) of any convex quadrangle (cf. (7))

$$\mathrm{conv}(P_1,P_2,P_3,P_4) = \Big\{(x,y) = \sum c_i P_i \, , \, c_i \ge 0 \, , \, \sum c_i = 1 \, , \, 1 \le i \le 4\Big\}$$

with $C(\mathcal{A}_n) = \left(\frac{2}{\pi}\log(n+1)+1\right)^2$ *and cardinality* n^2+2n+1 *(which becomes* n^2+n+1 *in the degenerate case of a triangle).*

Proof. First, by properties **(II)** and **(III)** of WAMs, tensor-product Chebyshev-Lobatto points, say $\mathcal{B}_n = \{(\xi_j, \xi_k) \, , \, 0 \le j,k \le n\}$, are a WAM for tensor-product polynomials of degree not greater than n, with mesh constant $C(\mathcal{B}_n) = \Lambda_n^2 \le \left(\frac{2}{\pi}\log(n+1)+1\right)^2$ (where Λ_n is the Lebesgue constant of univariate Chebyshev-Lobatto points, cf. [9]). Then, it is sufficient to observe that, for any polynomial $p \in \mathbb{P}_n^2$, $q(u,v) = p(\sigma(u,v))$ is a tensor product polynomial in $[-1,1]^2$, since the transformation σ is bilinear. We can conclude using the fact that σ is surjective, even in the degenerate case of a triangle, which entails that for every (x,y) in the quadrangle

$$|p(x,y)| = |q(u,v)| \le \Lambda_n^2 \, \|q\|_{\mathcal{B}_n} = \Lambda_n^2 \, \|p\|_{\mathcal{A}_n} \le \left(\frac{2}{\pi}\log(n+1)+1\right)^2 \|p\|_{\mathcal{A}_n} \, .$$

By bilinearity, any segment $u = const$ or $v = const$ in $[-1,1]^2$ is mapped onto a segment of the quadrangle. This entails that the WAM is an "oblique" grid and that the points on each segment, including the sides, are exactly its Chebyshev-Lobatto points; see Figure 1. Concerning the cardinality, observe that the transformation is bijective for nondegenerate convex quadrangles, whereas in the degenerate case of a triangle one side collapses into one vertex. □

We recall that these "oblique" Chebyshev-Lobatto grids are familiar in the context of quadrangle-based spectral and high-order element methods for PDEs (see, e.g., [12]). Indeed, they give the interpolation points for the construction of the trial functions via tensorial interpolation on the reference square (this basis is not polynomial in general, due to composition with the inverse mapping $\sigma^{-1}(x,y)$). What we are doing here is to reconsider Chebyshev-Lobatto grids in the context of *total degree* polynomial approximation over quadrangles, and in particular in the context of total degree Weakly Admissible Meshes, that are essentially a matter of polynomial inequalities.

4 Interpolation and cubature over polygons.

Given a *simple polygon* K, in view of Proposition 1 above and property **(IV)** of WAMs we can immediately construct a polygon WAM with $\mathcal{O}(n^2)$ points by

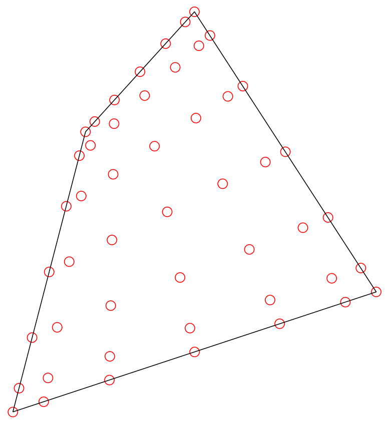

Figure 1: The $7 \times 7 = 49$ points of the Chebyshev-Lobatto WAM for degree $n = 6$ on a convex quadrangle.

finite union, once a *convex quadrangulation* of the polygon is at hand (where we can even admit the presence of some triangles). Quadrangulations are preferable with respect to triangulations in the present context, in order to reduce the cardinality of the resulting mesh. For the same reason we look for a coarse (possibly minimal), rather than for a fine quadrangulation.

We do not go into technical details of (convex) quadrangulation, which is a much less developed and understood problem than triangulation, and refer the reader e.g. to [8] and references therein for some insight into this difficult topic of computational geometry. We only observe that any simple polygon with k vertices can be partitioned into $k-2$ triangles, or into μ quadrangles (some of which possibly degenerating into triangles) with $(k-2)/2 \leq \mu \leq k-2$, where μ is often close to the lower bound. For example, any convex polygon is trivially partitioned into $(k-2)/2$ quadrangles for k even, or into $(k-3)/2$ quadrangles plus one triangle for k odd, simply by taking quadruples of consecutive vertices.

Once we have computed the array $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)$ corresponding to a Discrete Extremal Set at degree n from a polygon WAM, by Algorithm AFP or Algorithm DLP, we can interpolate any function by solving the Vandermonde-like system

$$V(\boldsymbol{\xi}, \boldsymbol{p})\boldsymbol{c} = \boldsymbol{f} \tag{9}$$

with a possibly much better conditioned matrix $V(\boldsymbol{\xi}, \boldsymbol{p}) = V(\boldsymbol{\xi}, \boldsymbol{q})T$ if iterated orthogonalization has been applied (cf. (5)-(6)).

Moreover, the corresponding array of algebraic cubature weights $\boldsymbol{w}$ with respect to a given measure μ on K, that are the weights of an exact formula on polynomials of degree not greater than n, can be computed by solving the linear system

$$V^t(\boldsymbol{\xi}, \boldsymbol{p})\boldsymbol{w} = \boldsymbol{m} \tag{10}$$

where $\boldsymbol{m}$ is the vector of *moments* of the polynomial basis

$$\boldsymbol{m} = \int_K \boldsymbol{p}(x, y)\, d\mu = T^t \int_K \boldsymbol{q}(x, y)\, d\mu \; . \tag{11}$$

In the case of Algorithm AFP, the cubature weights are a byproduct simply by taking $\boldsymbol{b} = \boldsymbol{m}$ as right-hand side, since the algorithm selects the points and at the same time computes the weights, by solving the relevant underdetermined linear system via QR factorization with column pivoting of the rectangular transpose Vandermonde matrix. On the other hand, with Algorithm DLP, it is sufficient to solve the couple of triangular systems $U^t\boldsymbol{z} = \boldsymbol{m}$, $L^t(:, 1 : N)\boldsymbol{w} = \boldsymbol{z}$. One important feature of Discrete Leja Points in this context is that they provide by construction a sequence of *nested* interpolatory cubature rules (for the concept of nested rule, see e.g. [23]). Concerning computation of the moments, we can resort for example to a recent algorithm [26], based on Green's integration formula and univariate Gaussian quadrature, that is able to compute exact moments of any polynomial basis on general simple polygons.

Numerical cubature over polygons has been traditionally thought as an application of polygon triangulation and cubature over triangles. In the recent literature, some new methods not based on triangulation have been proposed, with a special interest in the application to polygonal finite elements; cf. [15, 16, 21, 22, 26]. The present cubature formulas can be seen as a new approach to (non minimal) algebraic cubature based on polygon quadrangulation. They could be useful as starting guess for the computation of (near) minimal cubature formulas on polygons, using the nonlinear optimization algorithms discussed in [21, 28, 30].

4.1 Numerical results.

In order to measure the quality of Discrete Extremal Sets, we have computed numerically the norms of the corresponding interpolation operator and cubature functional, namely the Lebesgue constant and the sum of the cubature weights absolute values. Indeed, the cubature weights are not all positive, but the negative ones turn out to be few and relatively small.

We have implemented a Matlab code named POLYGINT (cf. [18]) that, given the polygon vertices counterclockwise ordered and the polynomial degree, computes Approximate Fekete Points and Discrete Leja Points from a Weakly Admissible Mesh obtained via polygon quadrangulation, together with the corresponding cubature weights. The polynomial basis for the Vandermonde matrix is the product Chebyshev basis of the minimal rectangle containing the polygon. Part of the code is based on the computational work made in [1, 17].

Polygon quadrangulation codes do not seem to be presently available in Matlab [20]. We have implemented a naive version of a convex quadrangulation algorithm, which subdivides any simple polygon $\mathcal{P}$ into convex polygons and then quadrangulates the convex elements (cf. [17, 18]).

The core of the algorithm is the following: for every possible concave angle formed by three consecutive vertices in counterclockwise order, say $\widehat{ABC}$, let Q the first point of intersection of the half-line through A and B with the polygon boundary (see Figure 2). The segment AQ subdivides the polygon into two new polygons, say $\mathcal{Q}$ and $\mathcal{S}$, by the auxiliary vertex Q. If we term ν the number of concave vertex angles of a polygon, then $\nu(\mathcal{Q}) + \nu(\mathcal{S}) < \nu(\mathcal{P})$. The algorithm is then applied to $\mathcal{Q}$ and $\mathcal{S}$, proceeding recursively until convex polygons are found (no concave angles) and then quadrangulated. It can be proved that if k is the number of vertices and ν the number of concave angles of the original polygon

$\mathcal{P}$, then the final quadrangulation has at most $\lceil \frac{k}{2} \rceil + \lfloor \frac{\nu+1}{2} \rfloor - 1$ quadrangles, of which at most $\nu + 1$ can be triangles (cf. [17]).

When $\nu \ll k$, the number of quadrangles is roughly half the number of triangles of a minimal triangulation. In very special cases ν can be close to k and the quadrangulation is not really competitive with a minimal triangulation, concerning the number of elements.

We present now some numerical tests on interpolation and cubature at Discrete Extremal Sets. All computations have been done by the POLYGINT code [18] in Matlab 7.6.0 on an Intel Core 2 Duo 2.13GHz Processor with 4Gb RAM.

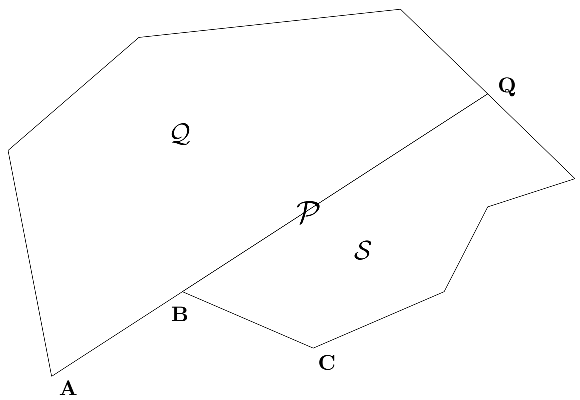

Figure 2: The basic step of polygon splitting by the quadrangulation algorithm.

In Figure 3 we show the polygon quadrangulation (top) and the Discrete Extremal Sets (bottom) computed at degree $n = 15$ by our algorithm (136 Approximate Fekete and Discrete Leja Points), on a concave test polygon in a shape of a hand, with 37 sides (obtained from the screen sampled hand of one of the authors, by piecewise linear interpolation). Our quadrangulation algorithm gives a subdivision of the polygon into 23 elements (quadrangles and triangles), whereas a minimal triangulation would consist of 35 triangles. This means that we are using an extraction WAM with approximately $23n^2$ instead of $35n^2$ points at each degree.

In Table 1 we report the norms of the interpolation operator and cubature functional at a sequence of degrees. In Table 2 we give the condition numbers of the Vandermonde-like matrices in the product Chebyshev basis of the minimal including rectangle, without and with the iterated basis orthogonalization, which strongly improves the conditioning. In Table 3 we show the CPU times of the computational process, where the linear algebra part soon becomes predominant with respect to the quadrangulation procedure.

Notice that the Lebesgue constants are much smaller than the theoretical estimate for Fekete points extracted from a WAM, namely $\Lambda_n \leq NC(\mathcal{A}_n) = (n+1)(n+2)\left(\frac{2}{\pi}\log^2(n+1)+1\right)^2/2$ (cf. property **(IV)** and Proposition 1). We see that Approximate Fekete Points give a better Lebesgue constant than Discrete Leja Points, and a slightly better norm of the cubature functional. This behavior has been confirmed by several other numerical experiments on various test polygons, not reported for brevity. Nevertheless, both turn out to be in practice good interpolation and cubature sets, as it is shown for example in Tables 4-5 by the errors on two test functions with different degree of regularity.

Table 1: Norms of the interpolation operator (Lebesgue constants) and of the cubature functional ($\|\boldsymbol{w}\|_1 = \sum |w_j|$) at Discrete Extremal Sets for the polygon in Figure 3. The size of the polygon area ($= \sum w_j$) is approximately 6.3e-2.

	pts	$n=3$ $N=10$	$n=6$ $N=28$	$n=9$ $N=55$	$n=12$ $N=91$	$n=15$ $N=136$	$n=18$ $N=190$
Λ_n	AFP	3.6	7.3	13.2	18.4	26.8	42.2
	DLP	7.0	10.2	26.0	35.1	44.6	78.7
$\|\boldsymbol{w}\|_1$	AFP	7.9e-2	6.8e-2	7.0e-2	6.9e-2	7.3e-2	7.0e-2
	DLP	7.0e-2	1.0e-1	1.0e-1	8.2e-2	1.2e-1	8.9e-2

Table 2: Condition numbers of Vandermonde-like matrices at Discrete Extremal Sets for the polygon in Figure 3 (without and with iterated orthogonalization), where $\boldsymbol{q}$ is the product Chebyshev basis of the minimal including rectangle.

	pts	$n=3$	$n=6$	$n=9$	$n=12$	$n=15$	$n=18$
$k_2(V(\boldsymbol{\xi}, \boldsymbol{q}))$	AFP	4.3e+1	1.4e+3	6.9e+4	4.4e+6	1.8e+8	8.4e+9
	DLP	6.8e+1	2.3e+3	1.2e+5	6.8e+6	1.7e+8	8.0e+9
$k_2(V(\boldsymbol{\xi}, \boldsymbol{q})T)$	AFP	2.3e+1	7.7e+1	1.8e+2	5.1e+2	1.1e+3	8.0e+2
	DLP	1.7e+1	1.6e+2	4.0e+2	6.0e+2	1.4e+3	3.0e+3

Table 3: CPU times (seconds) for the computation of points and weights for the polygon in Figure 3; the quadrangulation time is around 0.2 seconds.

pts	$n=3$	$n=6$	$n=9$	$n=12$	$n=15$	$n=18$
AFP	0.04	0.04	0.14	0.66	2.08	4.95
DLP	0.02	0.06	0.35	1.05	2.53	5.49

References

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

[2] R. Berman, S. Boucksom and D. Witt Nyström, Fekete points and convergence towards equilibrium measures on complex manifolds, arXiv:0907.2820 (Acta Math., to appear).

Table 4: Interpolation (absolute) and cubature (relative) errors for the function $f(x,y) = \cos(x+y)$ on the polygon in Figure 3; the reference value of the integral ($d\mu = dx\,dy$) is 4.08555342188148262749791 68e-2.

	pts	$n=3$	$n=6$	$n=9$	$n=12$	$n=15$	$n=18$
intp err	AFP	6.4e-5	6.6e-10	5.1e-15	1.3e-15	1.9e-15	1.9e-15
	DLP	8.1e-5	9.1e-10	7.6e-15	1.2e-15	1.6e-15	2.4e-15
cub err	AFP	4.1e-6	4.6e-12	1.0e-15	2.4e-15	2.0e-15	1.0e-15
	DLP	1.3e-7	6.5e-12	6.8e-16	2.7e-16	2.7e-15	2.9e-15

Table 5: As in Table 4 for the function $f(x,y) = ((x-0.45)^2 + (y-0.4)^2)^{3/2}$; the reference value of the integral is 1.59153824469955942379206 79e-4.

	pts	$n=3$	$n=6$	$n=9$	$n=12$	$n=15$	$n=18$
intp err	AFP	1.5e-3	1.2e-4	2.4e-5	1.3e-5	7.0e-6	3.5e-6
	DLP	2.9e-3	9.2e-5	3.9e-5	1.5e-5	9.3e-6	4.5e-6
cub err	AFP	5.4e-2	5.9e-3	3.8e-4	6.9e-5	3.4e-6	1.5e-5
	DLP	4.7e-2	5.4e-3	3.0e-4	1.7e-4	5.9e-5	8.0e-6

[3] L. Bos, J.-P. Calvi, N. Levenberg, A. Sommariva and M. Vianello, Geometric Weakly Admissible Meshes, Discrete Least Squares Approximation and Approximate Fekete Points, Math. Comp., published online January 19, 2011.

[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, N. Levenberg and S. Waldron, On the Spacing of Fekete Points for a Sphere, Ball or Simplex, Indag. Math. 19 (2008), 163–176.

[6] L. Bos, A. Sommariva and M. Vianello, Least-squares polynomial approximation on weakly admissible meshes: disk and triangle, J. Comput. Appl. Math. 235 (2010), 660–668.

[7] L. Bos, M.A. Taylor and B.A. Wingate, Tensor product Gauss-Lobatto points are Fekete points for the cube, Math. Comp. 70 (2001), 1543–1547.

[8] P. Bose, S. Ramaswami, G. Toussaint and A. Turki, Experimental Results on Quadrangulations of Sets of Fixed Points, Computer Aided Geometric Design 19 (2002), 533–552.

[9] L. Brutman, Lebesgue functions for polynomial interpolation—a survey, Ann. Numer. Math. 4 (1997), 111–127.

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

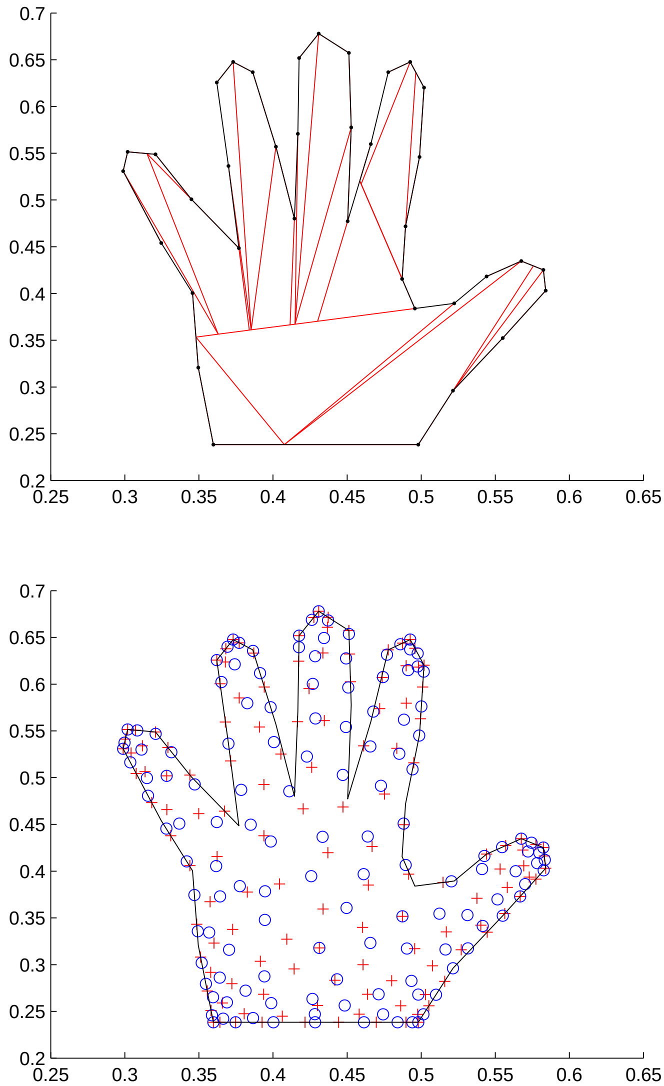

Figure 3: Quadrangulation (top) and Discrete Extremal Sets (bottom) of degree $n = 15$ for a hand shape polygon: 136 Approximate Fekete Points (+) and Discrete Leja Points (∘).

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

[12] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods. Evolution to Complex Geometries and Applications to Fluid Dynamics, Springer, Berlin, 2007.

[13] 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.

[14] S. De Marchi, M. Marchioro and A. Sommariva, Polynomial approximation and cubature at approximate Fekete and Leja points of the cylinder, preprint (available online at: www.math.unipd.it/∼marcov/CAApubl.html).

[15] C.-J. Li and C. Dagnino, An adaptive numerical integration algorithm for polygons, Appl. Numer. Math. 60 (2010), 165–175.

[16] 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.

[17] M. Gentile, An algorithm for convex quadrangulation of polygons, Laurea Thesis in Mathematics (Italian), University of Padova, 2010 (advisors A. Sommariva and M. Vianello).

[18] M. Gentile, A. Sommariva and M. Vianello, POLYGINT: a Matlab code for interpolation and cubature at Discrete Extremal Sets of polygons (available online at: www.math.unipd.it/∼marcov/CAAsoft.html).

[19] 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.

[20] MathWorks, MATLAB, http://www.mathworks.com/products/matlab and http://www.mathworks.com/matlabcentral/fileexchange.

[21] 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.

[22] S. Natarajan, S. Bordas and D.R. Mahapatra, Numerical integration over arbitrary polygonal domains based on Schwarz-Christoffel conformal mapping, Int. J. Numer. Meth. Engng. 80 (2009), 103–134.

[23] T.N.L. Patterson, Stratified nested and related quadrature rules, J. Comput. Appl. Math. 112 (1999), 243–251.

[24] E.B. Saff and V. Totik, Logarithmic Potentials with External Fields, Springer, 1997.

[25] I.H. Sloan and R.S. Womersley, Extremal systems of points and numerical integration on the sphere, Adv. Comput. Math. 21 (2004), 107–125.

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

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

[28] 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.

[29] M.A. Taylor, B.A. Wingate and R.E. Vincent, An algorithm for computing Fekete points in the triangle, SIAM J. Numer. Anal. 38 (2000), 1707–1720.

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