Computers and Mathematics with Applications 57 (2009) 1324–1336

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

# Computing approximate Fekete points by QR factorizations of Vandermonde matrices☆

Alvise Sommariva, Marco Vianello *

*Department of Pure and Applied Mathematics, University of Padova, Italy*

ARTICLE INFO

*Article history:*
Received 20 March 2008
Received in revised form 28 October 2008
Accepted 8 November 2008

*Keywords:*
Approximate Fekete points
Vandermonde matrices
Maximum volume submatrices
Greedy algorithm
QR factorization with column pivoting
Algebraic quadrature
Polynomial interpolation
Lebesgue constant
Admissible mesh

ABSTRACT

We propose a numerical method (implemented in Matlab) for computing approximate Fekete points on compact multivariate domains. It relies on the search of maximum volume submatrices of Vandermonde matrices computed on suitable discretization meshes, and uses a simple greedy algorithm based on QR factorization with column pivoting. The method gives also automatically an algebraic cubature formula, provided that the moments of the underlying polynomial basis are known. Numerical tests are presented for the interval and the square, which show that approximate Fekete points are well suited for polynomial interpolation and cubature.



## 1. Introduction

Let $\Omega \subset \mathbb{R}^d$ be a compact subset (or lower dimensional manifold). Given a polynomial basis for $\Pi_n^d(\Omega)$ (the subspace of $d$-variate polynomials of total degree $\leq n$ restricted to $\Omega$), say

$$\text{span}(p_j)_{1\leq j\leq N} = \Pi_n^d(\Omega), \qquad N = N(n) := \dim(\Pi_n^d(\Omega)), \tag{1}$$

and a sufficiently large and dense discretization of $\Omega$

$$X = \{x_i\} \subset \Omega, \quad 1 \leq i \leq M,\ M > N, \tag{2}$$

we can construct the rectangular Vandermonde matrix

$$V = V_n(x_1, \ldots, x_M) = (v_{ij}) := (p_j(x_i)) \in \mathbb{R}^{M\times N}. \tag{3}$$

A quadrature formula of algebraic degree of exactness $n$ for a given measure $\mu$ on $\Omega$ can be obtained by solving the underdetermined linear system of the quadrature weights

$$\sum_{i=1}^{M} w_i\, p_j(x_i) = \int_\Omega p_j(x)\mathrm{d}\mu, \quad 1 \leq j \leq N \tag{4}$$

☆ Work supported by the "ex-60%" funds of the University of Padova, and by the INdAM GNCS. This work is published despite the drastic reduction of public funding for universities and research pursued by the Italian government (see the article "Cut-throat savings", Nature 455, October 2008, http://www.nature.com/nature/journal/v455/n7215/full/455835b.html).

* Corresponding author.
*E-mail address:* marcov@math.unipd.it (M. Vianello).

0898-1221/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2008.11.011

that is in matrix form

$$V^t w = m, \qquad m = \{m_j\} = \left\{ \int_\Omega p_j(x) \mathrm{d}\mu \right\}, \quad 1 \le j \le N, \tag{5}$$

provided that the "moments" $\{m_j\}$ are explicitly known or computable (cf., e.g., [1,2] for the computation of polynomial moments over nonstandard domains). We observe that in the numerical literature there is no universal agreement on the terminology on Vandermonde matrices, often $V^t$ (in our notation) is called the Vandermonde matrix; see, e.g., [3].

The solution of such a system by a standard SVD approach (cf. [3,4]) would then give in general a vector of $M$ nonzero weights, and thus a quadrature formula which uses all the original discretization points; this has been confirmed by all our numerical experiments, where zero or nearly zero weights (compared to the others) do not appear, indeed all the weights have the same size. On the contrary, if $V^t$ has full rank, its solution by the standard Matlab backslash "\" solver for linear systems (cf. [5]) gives only $N$ nonzero weights, and thus also an automatic selection of the relevant $N$ quadrature nodes and weights. In a Matlab-like notation we can write:

**Algorithm 1** (*Approximate Fekete points, Full Rank $V^t$*).

- $w = V^t \setminus m$; ind = find$(w \neq 0)$;
- $X_* = X(\mathrm{ind})$; $w_* = w(\mathrm{ind})$; $V_* = V(\mathrm{ind}, :)$;

where ind $= (i_1, \ldots, i_N)$, that is we get the two arrays of length $N$

$$X_* = \{x_{i_1}, \ldots, x_{i_N}\}, \qquad w_* = \{w_{i_1}, \ldots, w_{i_N}\}, \tag{6}$$

which generate the quadrature formula

$$\int_\Omega f(x) \mathrm{d}\mu \approx \sum_{k=1}^{N} w_{i_k} f(x_{i_k}), \quad f \in C(\Omega). \tag{7}$$

Moreover, we also extract a nonsingular Vandermonde submatrix $V_*$ (corresponding to the selected points), which can be useful for polynomial interpolation. When $\mathrm{rank}(V^t) < N$, Algorithm 1 fails to extract the correct number of points. This means that polynomial interpolation at that degree is not possible by the extraction procedure (a remedy for this situation is discussed in Section 3). Computing quadrature weights from moments via square Vandermonde matrices is a well-known and developed approach in the one-dimensional case (see, e.g., [6]), whereas the general and multidimensional extraction procedures just sketched (based on rectangular Vandermonde matrices) seems in some respect new.

Our numerical experiments in the interval and in the square show that, when suitable polynomial bases are used, this approach gives a good (stable and convergent) quadrature formula, and in addition the extracted quadrature nodes are good polynomial interpolation points (slow growth of the Lebesgue constant).

This is essentially due to the implementation of the Matlab backslash command for underdetermined linear systems, which is based on the QR factorization algorithm with column pivoting, firstly proposed by Businger and Golub in 1965 (cf. [7]). The backslash command for rectangular matrices uses the LAPACK routine DGEQP3, see the "mldivide" command page in [5,8].

The result is that such points are approximate Fekete points, that is points computed by "trying to maximize" the Vandermonde determinant absolute value, as we shall discuss in the next sections. Our work is mainly of computational kind, for a deep discussion about the theoretical issues of the present approach in the one-dimensional case we refer the reader to the work in progress [9].

## 2. Approximate Fekete points in the interval

In order to show the potentialities of the method, we present in Table 1 below some relevant parameters concerning quadrature and interpolation in the one-dimensional case, $\Omega = [-1, 1]$ and $\mathrm{d}\mu = \mathrm{d}x$. The nodes are extracted from a uniform grid of 5000 points (see Remark 1) at a sequence of degrees, $n = 10, 20, \ldots, 60$, with three different polynomial bases (the monomial, the Legendre and the Chebyshev basis). The parameters (given with two or three significant figures) are the spectral condition number of the transpose rectangular Vandermonde matrix, the euclidean norm of the weights' system residual (say $\|\mathrm{res}(w)\|_2 = \|m - V^t w\|_2$), the sum of the weights' absolute values (a measure of the quadrature stability, cf. [10]), the Lebesgue constant $\Lambda_n$ (a measure of the interpolation stability, cf. [11]: such a quantity is evaluated numerically on a very large set of control points). Concerning quadrature, the required moments are known analytically, in particular by orthogonality the integrals of the Legendre basis polynomials are all vanishing except at degree zero (cf., e.g., [12] for the Chebyshev basis).

We can see that both the orthogonal bases give very good results, the best in terms of Lebesgue constant being obtained with the Chebyshev basis. On the contrary, the monomial basis suffers from ill-conditioning of Vandermonde matrices, which at higher degrees become even rank-deficient, i.e. $\mathrm{rank}(V^t) < N$, so that as observed above Algorithm 1 fails to extract the correct number of points. This means that polynomial interpolation at that degree is not possible by the extraction

**Table 1**
Relevant parameters for the extracted points, $\Omega = [-1, 1]$ (where $(+)$ means that the weights are all positive, and $*$ means that Algorithm 1 fails to extract $N = n + 1$ points due to rank-deficiency).

| Basis | | $n = 10$ | $n = 20$ | $n = 30$ | $n = 40$ | $n = 50$ | $n = 60$ |
|---|---|---|---|---|---|---|---|
| Mon | $\mathrm{cond}(V^t)$ | 3.1E+03 | 1.8E+07 | 1.1E+11 | 6.7E+14 | 1.7E+16 | 3.6E+16 |
| | $\|res(w)\|_2$ | 5.5E−16 | 1.4E−15 | 5.1E−12 | 2.8E−10 | 3.1E−10 | 2.0E−10 |
| | $\sum \|w_{i_k}\|$ | 2.00(+) | 2.00(+) | 5.31 | 5.47 | 93.6 | 56.7 |
| | $\Lambda_n$ | 5.33 | 5.06 | ∗(30pts) | ∗(30pts) | ∗(32pts) | ∗(35pts) |
| Leg | $\mathrm{cond}(V^t)$ | 4.6E+00 | 6.4E+00 | 7.8E+00 | 9.0E+00 | 1.0E+01 | 1.1E+01 |
| | $\|res(w)\|_2$ | 6.6E−16 | 7.0E−16 | 1.1E−15 | 2.3E−15 | 1.4E−15 | 2.0E−15 |
| | $\sum \|w_{i_k}\|$ | 2.00(+) | 2.00(+) | 2.01 | 2.05 | 2.00(+) | 2.00(+) |
| | $\Lambda_n$ | 2.74 | 5.94 | 7.11 | 9.59 | 10.9 | 12.4 |
| Cheb | $\mathrm{cond}(V^t)$ | 3.7E+00 | 5.0E+00 | 6.0E+00 | 6.7E+00 | 7.1E+00 | 7.5E+00 |
| | $\|\mathrm{res}(w)\|_2$ | 1.2E−15 | 1.4E−15 | 1.6E−15 | 1.8E−15 | 1.9E−15 | 2.1E−15 |
| | $\sum \|w_{i_k}\|$ | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) |
| | $\Lambda_n$ | 2.27 | 2.79 | 3.13 | 3.40 | 3.58 | 3.80 |

procedure. Nevertheless, a quadrature formula can be constructed with a number of points (in parentheses) that is less than the dimension of the polynomial space of "exactness", and which still gives acceptable results (see also Table 5 below). Observe that we could have computed the Lebesgue constant also in the cases of failure using the available nodes, but we have avoided this since it makes sense only in one dimension (where any natural number can be the dimension of a polynomial subspace).

The good behavior of the points selected by Algorithm 1, for algebraic quadrature and for polynomial interpolation (at least when suitable polynomial bases are used), is directly related with the features of the Matlab "backslash" linear solver. In fact, in the case of underdetermined systems, such a solver performs a QR factorization with column pivoting of the matrix (cf. [4,5,7]). In practice, this corresponds to a special QR factorization of the transpose Vandermonde submatrix $V_*^t \in \mathbb{R}^{N\times N}$, that is

$$(V^t P)(:, 1 : N) = V_*^t = QR, \tag{8}$$

where $Q$ is orthogonal, $R$ upper triangular with $|r_{11}| \geq |r_{22}| \geq \cdots \geq |r_{NN}|$, and $P$ a permutation matrix. More precisely, since inside a QR process with stepwise selection of the columns, we can see the diagonal element of $R$ produced at a given step as a function of the matrix columns involved so far,

$$r_{kk} = r_{kk}(\gamma_1, \ldots, \gamma_k), \quad 0 \leq k \leq N, \qquad \gamma_j \in \left\{col_1(V^t), \ldots, col_M(V^t)\right\} \subset \mathbb{R}^N, \tag{9}$$

it can be shown that the QR algorithm with column pivoting acts in such a way to maximize $r_{kk}$ as a function of the vector variable $\gamma_k$ (the vector variables $\gamma_1, \ldots, \gamma_{k-1}$ having been fixed by the previous steps); cf., e.g., [3, Section 2.7.3]. In other words, since

$$|\det(V_*^t)| = |\det(V_*)| = \prod_{k=1}^{N} |r_{kk}|, \tag{10}$$

all the process can be re-interpreted as a heuristic optimization of the extracted Vandermonde determinant (as a function of $N$ of the $M$ original discretization points), based on sequential componentwise maximization of the factors.

An equivalent but more "geometric" interpretation of Algorithm 1 is that related with the notion of volume of submatrices. Indeed, it can be shown that the QR factorization (8) (the core of the algorithm) is an implementation of the standard "greedy" approximation algorithm for selecting $N$ columns with maximal associated volume, which can be sketched as follows:

**Algorithm greedy** (*Max Volume Submatrix of* $A \in \mathbb{R}^{N\times M}$, $M > N$).

- ind = [ ] ;
- for $k = 1, \ldots, N$
  - "select the largest norm column $col_{i_k}(A)$"; ind = [ind, $i_k$];
  - "remove from every column of $A$ its orthogonal projection onto $col_{i_k}$;
  end;

see, e.g., [9,13] and the references therein. It is worth stressing that (8) acts only on the matrix and thus the mere selection of the points in Algorithm 1 could be done with any right-hand side in the system (i.e., it is independent of the specific quadrature problem).

Observe that such a discrete nonlinear optimization problem is known to be NP-Hard (cf. [13]), and thus heuristic/stochastic methods are mandatory. The strength of Algorithm 1 is that it gives good results in practice, by using only basic optimized tools of numerical linear algebra. We stress that the use of the commercial package Matlab for this

**Table 2**
Absolute value of the Vandermonde determinants (in the Chebyshev basis) for different families of points, $\Omega = [-1, 1]$; $*$ indicates algorithm failure due to rank-deficiency.

| Points | $n = 10$ | $n = 20$ | $n = 30$ | $n = 40$ | $n = 50$ | $n = 60$ |
|---|---|---|---|---|---|---|
| Eq spaced | 8.4E+02 | 2.0E+03 | 6.7E−01 | 1.3E−08 | 1.1E−20 | 2.2E−37 |
| Approx Fek | | | | | | |
| Basis mon | 9.6E+03 | 7.3E+10 | * | * | * | * |
| Basis Leg | 2.4E+04 | 6.5E+10 | 1.7E+18 | 2.0E+26 | 1.0E+35 | 2.2E+44 |
| Basis Cheb | 3.1E+04 | 1.5E+11 | 8.4E+18 | 2.3E+27 | 2.0E+36 | 5.6E+45 |
| True Fek | 3.1E+04 | 1.5E+11 | 8.6E+18 | 2.5E+27 | 2.4E+36 | 6.2E+45 |
| ChebLob | 2.8E+04 | 1.3E+11 | 6.8E+18 | 2.0E+27 | 1.8E+36 | 4.5E+45 |
| GaussCheb | 1.7E+04 | 7.5E+10 | 4.0E+18 | 1.1E+27 | 1.0E+36 | 2.7E+45 |

problem is user-friendly and practical due to its wide diffusion, but the implementation can be done by other open-source libraries and computing systems, e.g. using directly LAPACK subroutines; cf. [8,14] and the references therein.

Points that maximize the Vandermonde volume in the continuum, the so-called Fekete (or extremal) points, are important in polynomial interpolation (see, e.g., [15,16]). This stems directly from the representation of the Lagrange cardinal polynomials for a given unisolvent set at degree $n$, say $\{\xi_1, \ldots, \xi_N\}$, as the ratio of two Vandermonde determinants

$$L_{\xi_i}(x) = \frac{\det(V_n(\xi_1, \ldots, \xi_{i-1}, x, \xi_{i+1}, \ldots, \xi_N))}{\det(V_n(\xi_1, \ldots, \xi_{i-1}, \xi_i, \xi_{i+1}, \ldots, \xi_N))}, \qquad L_{\xi_i}(\xi_k) = \delta_{ik}, \tag{11}$$

(cf. (3) for the definition of $V_n$), from which it is clear that at a subset of $\Omega$ which maximizes the absolute value of the Vandermonde determinant, say $\mathcal{F}_n = \{\phi_1, \ldots, \phi_N\}$, we have that the Lebesgue constant (the norm of the interpolation operator) is bounded by $N$

$$\|L_{\phi_i}\|_\infty = 1, \quad 1 \le i \le N \Longrightarrow \Lambda_n := \max_{x\in\Omega} \sum_{i=1}^{N} |L_{\phi_i}(x)| \le N. \tag{12}$$

Such a rough estimate already shows that Fekete points are good interpolation points. Moreover, they can be also near-optimal interpolation points, as it happens in the one-dimensional case, where they are known to be the Gauss–Lobatto points by a classical result of Fejér and to have a Lebesgue constant growing like $\mathcal{O}(\log n)$ (cf. [17]). Much less is known in higher dimension, see [18] and the references therein.

The numerical computation of high-degree Fekete points on a given $d$-dimensional compact subset is a hard large-scale problem, since it corresponds to the optimization of a nonlinear function with $2N$ variables (recall that $N \sim n^d/d!$). Indeed even in important two-dimensional instances, like the triangle which is relevant for the application to spectral element methods for PDEs, Fekete points have been computed only up to relatively small degrees; cf., e.g., [19–21] and the references therein. A big effort has been made to compute Fekete (or extremal) points on the sphere, in view of their importance in applications, by methods that need large-scale computational resources, cf. [22] and the references therein.

On the other hand, also the computation of good points for algebraic quadrature over $d$-dimensional compact subsets, especially for the so-called minimal quadrature formulas, is a substantially open problem with several important applications, whose direct numerical solution again involves large-scale nonlinear problems; cf., e.g., [23,24] and the references therein.

Our numerical experiments have shown that Algorithm 1 gives a reasonable compromise between quality of the quadrature/interpolation points, and computational cost. As sketched above, the method is related with the maximization of Vandermonde volumes, and thus we can call the produced points "approximate Fekete points". The following tables give more evidence in this direction, by comparing the Vandermonde volume (Table 2) and the Lebesgue constant (Table 3) of the one-dimensional Fekete points with that of equally spaced points, of the three families of approximate Fekete points of Table 1, and of the Gauss–Chebyshev and Chebyshev–Lobatto points which are known to be excellent interpolation points (the Vandermonde matrix is computed in the Chebyshev basis in all instances).

In Table 4, we report the distance in $\|\cdot\|_\infty$ of different arrays of approximate Fekete points from the true Fekete points (ordered arrays). The superiority of the orthogonal bases is confirmed also in terms of these parameters: in particular, the approximate Fekete points obtained by the Chebyshev basis are in all respects the closest to the true Fekete points.

It is worth stressing the fact that, at least in the complex case (polynomial interpolation in $\mathbb{C}$), there is a sound theoretical basis for the connection of our approximate discrete Fekete points with the continuum Fekete points, as is made clear by the following key result (cf. [9]) which applies in particular to the case of the interval.

**Theorem 1** (*Bos and Levenberg, [9]*). *Suppose that $\Omega \subset \mathbb{C}$ is a continuum (i.e. compact and connected, not a single point). Suppose further that $X_n \subset \Omega$, $n = 1, 2, \ldots$, are discrete subsets of $\Omega$ such that for all $x \in \Omega$*

$$\min_{y\in X_n} |x - y| \le \phi(n), \quad \textit{where } \lim_{n\to\infty} n^2\phi(n) = 0. \tag{13}$$

*(For example, if $\Omega = [-1, 1]$, $X_n$ consisting of order $n^{2+\varepsilon}$, $\varepsilon > 0$, equally spaced points would suffice).*

**Table 3**
Comparison of the Lebesgue constants for different families of points, $\Omega = [-1, 1]$; $*$ indicates algorithm failure due to rank-deficiency.

| Points | $n = 10$ | $n = 20$ | $n = 30$ | $n = 40$ | $n = 50$ | $n = 60$ |
|---|---|---|---|---|---|---|
| Eq spaced | 29.9 | 1.10E+04 | 6.60E+06 | 4.05E+08 | 7.34E+09 | 1.24E+10 |
| Approx Fek | | | | | | |
| Basis mon | 5.33 | 5.06 | $*$ | $*$ | $*$ | $*$ |
| Basis Leg | 2.74 | 5.94 | 7.11 | 8.59 | 10.9 | 12.4 |
| Basis Cheb | 2.27 | 2.79 | 3.13 | 3.40 | 3.58 | 3.80 |
| True Fek | 2.18 | 2.61 | 2.86 | 3.04 | 3.18 | 3.30 |
| ChebLob | 2.42 | 2.87 | 3.13 | 3.31 | 3.45 | 3.57 |
| GaussCheb | 2.49 | 2.90 | 3.15 | 3.33 | 3.47 | 3.58 |

**Table 4**
Distances in $\|\cdot\|_\infty$ of the approximate from the true Fekete points' array, $\Omega = [-1, 1]$ (ordered arrays); $*$ indicates algorithm failure due to rank-deficiency.

| Points | $n = 10$ | $n = 20$ | $n = 30$ | $n = 40$ | $n = 50$ | $n = 60$ |
|---|---|---|---|---|---|---|
| Approx Fek | | | | | | |
| Basis mon | 1.3E−01 | 4.3E−02 | $*$ | $*$ | $*$ | $*$ |
| Basis Leg | 5.9E−02 | 6.1E−02 | 5.8E−02 | 4.2E−02 | 3.0E−02 | 2.4E−02 |
| Basis Cheb | 9.1E−03 | 6.6E−03 | 6.1E−03 | 4.8E−03 | 4.0E−03 | 3.6E−03 |

**Table 5**
Quadrature and interpolation errors for the Runge function; $*$ indicates algorithm failure due to rank-deficiency.

| Points | | $n = 10$ | $n = 20$ | $n = 30$ | $n = 40$ | $n = 50$ | $n = 60$ |
|---|---|---|---|---|---|---|---|
| Approx Fek | | | | | | | |
| Basis mon | Quadr | 1.9E−02 | 2.0E−04 | 3.5E−04 | 8.2E−05 | 4.9E−04 | 4.8E−04 |
| | Interp | 1.9E+00 | 1.5E−01 | $*$ | $*$ | $*$ | $*$ |
| Basis Leg | Quadr | 2.7E−03 | 1.6E−03 | 3.9E−05 | 6.8E−06 | 3.8E−07 | 6.7E−08 |
| | Interp | 1.6E+00 | 1.8E−01 | 1.5E−02 | 1.3E−03 | 9.8E−05 | 1.1E−05 |
| Basis Cheb | Quadr | 9.6E−03 | 6.7E−05 | 6.8E−07 | 1.2E−07 | 2.1E−09 | 6.8E−10 |
| | Interp | 1.7E+00 | 1.3E−01 | 1.1E−02 | 9.6E−04 | 8.1E−05 | 6.8E−06 |
| True Fek | Quadr | 8.9E−03 | 6.2E−05 | 4.4E−07 | 3.1E−09 | 2.2E−11 | 1.6E−13 |
| | Interp | 1.6E+00 | 1.3E−01 | 1.1E−02 | 9.5E−04 | 7.8E−05 | 6.7E−06 |

*Then, independently of the bases used, Algorithm 1 will generate sets of approximate Fekete points that have the same asymptotic distribution (see [25]) as do the true Fekete points. Specifically, the positive measure of total mass 1 obtained by assigning to every point a mass of $1/(n + 1)$ converges weakly to the equilibrium measure for $\Omega$.*

**Remark 1.** It is worth recalling that many other families of points have the same asymptotic distribution in the case of the interval, $\Omega = [-1, 1]$, e.g. the zeros of Jacobi polynomials. The property that the approximate Fekete points array converges in $\|\cdot\|_\infty$ to the true Fekete points is stronger than the weak-* convergence given by Theorem 1, and has only some numerical evidence such as that given in Table 4.

Notice that the number $M = 5000$ of discretization points used in the numerical experiments on the interval, is consistent with the assumptions of Theorem 1 up to the highest degree ($n^2 = 60^2 = 3600$). On the other hand, it is also consistent with the spacing of the true Fekete points (the minimum of pairwise distances) in complex compact sets of unit capacity, which is bounded from below by $2/(en^2)$ for $n \geq 4$, cf. [25]. The number of extraction points could be tuned to the degree, namely one could compute approximate Fekete points in $[-1, 1]$ at degree $n$ from a uniform grid with stepsize $1/(en^2)$. This has been tried in our numerical experiments, obtaining results close to those displayed in Tables 1–5.

We conclude this section by showing a numerical test (Table 5), where the quadrature and interpolation errors of the Runge function $f(x) = 1/(1 + 16x^2)$ at approximate Fekete points of Table 1 are compared with those at the true Fekete points. The interpolating polynomial has been computed by solving the corresponding square Vandermonde system with standard Gaussian elimination (again, the "\" command in Matlab).

As expected, the points obtained via the Chebyshev–Vandermonde matrix are excellent quadrature/interpolation points. Notice that the points corresponding to the monomial basis even in the presence of severe ill-conditioning and rank-deficiency are able to give acceptable quadrature results, but with an observed error stalling around $\mathcal{O}(10^{-4})$.

## 3. Iterative refinement

The approximate Fekete points computed by Algorithm 1 depend on the polynomial basis adopted, as is clear from the previous tables (even though, in the one-dimensional case, all these points should have asymptotically the same distribution as do the true Fekete points, in view of Theorem 1).

Using "wrong" bases, especially for the interpolation problem, leads to "bad" sets of points, due to the extreme ill-conditioning of the Vandermonde matrices, or even to computational failure since such matrices can become rank-deficient. As it is well known in the one-dimensional case, the problem of ill-conditioning is typical of the monomial basis. We recall, for example, that in view of the results of [26], the spectral conditioning of $V^t$ for the monomial basis when $M \gg N = n+1$ is expected to be close to the square root of the conditioning of the Hilbert matrix of order $N$. The ill-conditioning can be attenuated or even eliminated by resorting to bases of orthogonal polynomials; cf., e.g., [3,27] and the references therein. This behavior is also evident in Table 1.

The following iterative refinement algorithm, based on successive changes of basis by QR factorizations of the (nontranspose) Vandermonde matrices (with Q rectangular orthogonal and R square upper triangular), tries to give a computational solution to the problem of the basis choice. This will be particularly relevant in dimension greater than 1, where bases of orthogonal polynomials are not known explicitly for general domains.

**Algorithm 2** (*Approximate Fekete Points by Iterative Refinement*).

- $V_0 = V$ ; $P_0 = I$ ;
- for $k = 0, \ldots, s-1$
  $V_k = Q_k R_k$; $U_k = \mathrm{inv}(R_k)$; $V_{k+1} = V_k U_k$ ; $P_{k+1} = P_k U_k$ ;
  end;
- $\mu = P_s^t m$; $w = V_s^t \setminus \mu$; $\mathrm{ind} = \mathrm{find}(w \neq 0)$ ;
- $X_* = X(\mathrm{ind})$; $w_* = w(\mathrm{ind})$; $V_* = V_s(\mathrm{ind}, :)$;

We stress that multiplication by $U_k$ corresponds to a change of basis in the Vandermonde matrix, in such a way that starting from a polynomial basis $p = (p_1, \ldots, p_N)$ the final basis will be $q^{(s)} = (q_1^{(s)}, \ldots, q_N^{(s)}) = pP_s$ (the column vector of moments must then be transformed into $\mu = P_s^t m$).

If $V_0 = V$ is not severely ill-conditioned, then $Q_0$ is numerically orthogonal and $V_1 = V_0 U_0$ is close to $Q_0$, that is the basis $q^{(1)} = pP_1$ is (numerically) orthogonal with respect to a discrete inner product defined on the original $M$ discretization points of $\Omega$

$$\langle q_j^{(1)}, q_h^{(1)} \rangle := \sum_{i=1}^{M} q_j^{(1)}(x_i)\, q_h^{(1)}(x_i) = (V_1^t V_1)_{jh} \approx (Q_0^t Q_0)_{jh} = 0, \quad j \neq h. \tag{14}$$

If $V_0$ is severely ill-conditioned, then $Q_0$ might be far from orthogonality, and in addition since also $R_0$ is ill-conditioned then $V_1 = V_0 U_0$ is not close to $Q_0$. Nevertheless, $Q_0$ and even $V_1$ are much better conditioned than $V_0$. When $V_0$ is numerically full rank (cf. [28]), the second iteration gives $Q_1$ orthogonal up to machine precision (the rule of "twice is enough", cf. [28]), and $V_2$ near-orthogonal or at least sufficiently well conditioned to apply successfully Algorithm 1. In practice, however, often one iteration is sufficient.

Algorithm 2 can work even when $V_0^t$ is numerically rank-deficient (see the asterisks in Table 6 for $iter = 0$), since after some iterations we can get full rank and then another one or two iterations usually suffice as just described. The functional interpretation of Algorithm 2 is that the basis $q^{(k)} = pP_k$ approaches a sort of discrete orthogonality at the original $M$ discretization points of $\Omega$.

In Tables 6–8, the effect of iterative refinement is shown on the three families of points of Table 1. Observe that 1–2 iterations give a substantial improvement of the quality of the extracted points, for both the monomial and the Legendre basis (beyond, a stalling of the stability parameters is observed). On the contrary, a slight worsening appears with the Chebyshev basis concerning the Lebesgue constant (but only on the second significant figure), since this is already an excellent basis for the extraction process, whereas the iterative refinement tends to produce the same discrete orthonormal basis (triangular like the starting bases). Indeed, unless the initial conditioning is too severe as with the monomial basis at high degrees, we can see that the final Lebesgue constants are (nearly) the same for all the bases, and increase very slowly (compare again with the true Fekete points).

In order to appreciate the quality of the approximate Fekete points obtained after the iterative refinement process, in Table 9 we compare again the quadrature and interpolation errors on the Runge test function. The interpolation matrix is $V_* = V_s(\mathrm{ind}, :) = V(\mathrm{ind}, :)P_s$, cf. Algorithm 2. The performance is now very good also with the Legendre basis at all degrees, and with the monomial basis at low degrees. With the monomial basis at higher degrees, where we have a severe ill-conditioning of the original Vandermonde matrix (which entails also a severe ill-conditioning of the transformation matrix $P_s$), we can observe an error stalling/worsening as in Table 5 (no refinement), but remaining in any case 2–3 orders of magnitude below the unrefined case.

## 4. Towards multivariate Fekete points

The computation of approximate Fekete points in multivariate instances is a challenging problem. In principle, Algorithms 1 and 2 can be applied in any compact domain, as soon as a suitable basis and initial set of points have been chosen. So, the basic questions are: given a multidimensional compact domain, which could be a reasonable distribution of points? and which could be a reasonable basis to work with? The answers depend strongly on the geometry of the domain.

**Table 6**
Iterative refinement of the approximate Fekete points extracted with the monomial basis, $\Omega = [-1, 1]$; $*$ indicates algorithm failure due to rank-deficiency.

| | iter | $n = 10$ | $n = 20$ | $n = 30$ | $n = 40$ | $n = 50$ | $n = 60$ |
|---|---|---|---|---|---|---|---|
| $\mathrm{cond}(V^t)$ | 0 | 3.1E+03 | 1.8E+07 | 1.1E+11 | 6.7E+14 | 1.7E+16 | 3.6E+16 |
| | 1 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.1E+09 | 3.3E+12 |
| | 2 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 |
| | 3 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 |
| $\|\mathrm{res}(w)\|_2$ | 0 | 5.5E−16 | 1.4E−15 | 5.1E−12 | 2.8E−10 | 3.1E−10 | 2.0E−10 |
| | 1 | 5.9E−16 | 1.1E−17 | 2.2E−17 | 2.3E−17 | 2.5E−14 | 7.3E−13 |
| | 2 | 7.5E−16 | 1.9E−17 | 2.1E−17 | 2.4E−17 | 2.9E−17 | 4.7E−17 |
| | 3 | 2.1E−16 | 4.7E−18 | 2.0E−17 | 1.7E−17 | 3.5E−17 | 3.5E−17 |
| $\sum \lvert w_{i_k}\rvert$ | 0 | 2.00(+) | 2.00(+) | 5.31 | 5.47 | 93.6 | 56.7 |
| | 1 | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 24.0 | 22.0 |
| | 2 | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.17 | 2.23 |
| | 3 | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.13 | 2.34 |
| $\Lambda_n$ | 0 | 5.33 | 5.06 | $*$ | $*$ | $*$ | $*$ |
| | 1 | 2.38 | 2.93 | 3.29 | 3.54 | 7.56 | 10.2 |
| | 2 | 2.38 | 2.93 | 3.29 | 3.54 | 5.49 | 9.69 |
| | 3 | 2.38 | 2.93 | 3.29 | 3.54 | 5.49 | 9.69 |
| True Fek $\Lambda_n$ | | 2.18 | 2.61 | 2.86 | 3.04 | 3.18 | 3.30 |

**Table 7**
Iterative refinement of the approximate Fekete points extracted with the Legendre basis, $\Omega = [-1, 1]$.

| | iter | $n = 10$ | $n = 20$ | $n = 30$ | $n = 40$ | $n = 50$ | $n = 60$ |
|---|---|---|---|---|---|---|---|
| $\mathrm{cond}(V^t)$ | 0 | 4.6E+00 | 6.4E+00 | 7.8E+00 | 9.0E+00 | 1.0E+01 | 1.1E+01 |
| | 1 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 |
| | 2 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 |
| $\|\mathrm{res}(w)\|_2$ | 0 | 6.6E−16 | 7.0E−16 | 1.1E−15 | 2.3E−15 | 1.4E−15 | 2.0E−15 |
| | 1 | 1.4E−17 | 1.7E−17 | 2.3E−17 | 2.8E−17 | 2.8E−17 | 3.6E−17 |
| | 2 | 1.4E−17 | 2.0E−17 | 2.1E−17 | 2.8E−17 | 2.2E−17 | 4.2E−17 |
| $\sum \lvert w_{i_k}\rvert$ | 0 | 2.00(+) | 2.00(+) | 2.01 | 2.05 | 2.00(+) | 2.00(+) |
| | 1 | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) |
| | 2 | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) |
| $\Lambda_n$ | 0 | 2.74 | 5.94 | 7.11 | 9.59 | 10.9 | 12.4 |
| | 1 | 2.38 | 2.93 | 3.29 | 3.54 | 3.72 | 3.90 |
| | 2 | 2.38 | 2.93 | 3.29 | 3.54 | 3.72 | 3.90 |
| True Fek $\Lambda_n$ | | 2.18 | 2.61 | 2.86 | 3.04 | 3.18 | 3.30 |

**Table 8**
Iterative refinement of the approximate Fekete points extracted with the Chebyshev basis, $\Omega = [-1, 1]$.

| | iter | $n = 10$ | $n = 20$ | $n = 30$ | $n = 40$ | $n = 50$ | $n = 60$ |
|---|---|---|---|---|---|---|---|
| $\mathrm{cond}(V^t)$ | 0 | 3.7E+00 | 5.0E+00 | 6.0E+00 | 6.7E+00 | 7.1E+00 | 7.5E+00 |
| | 1 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 |
| | 2 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 |
| $\|\mathrm{res}(w)\|_2$ | 0 | 1.2E−15 | 1.4E−15 | 1.6E−15 | 1.8E−15 | 1.9E−15 | 2.1E−15 |
| | 1 | 1.0E−17 | 2.0E−17 | 2.0E−17 | 2.4E−17 | 2.8E−17 | 2.7E−17 |
| | 2 | 7.0E−18 | 1.8E−17 | 2.2E−17 | 2.7E−17 | 2.3E−17 | 2.9E−17 |
| $\sum \lvert w_{i_k}\rvert$ | 0 | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) |
| | 1 | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) |
| | 2 | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) | 2.00(+) |
| $\Lambda_n$ | 0 | 2.27 | 2.79 | 3.13 | 3.40 | 3.58 | 3.80 |
| | 1 | 2.38 | 2.93 | 3.29 | 3.54 | 3.72 | 3.90 |
| | 2 | 2.38 | 2.93 | 3.29 | 3.54 | 3.72 | 3.90 |
| True Fek $\Lambda_n$ | | 2.18 | 2.61 | 2.86 | 3.04 | 3.18 | 3.30 |

Concerning the basis, the computational experience and the univariate theory (cf. [3,27]) show that a better conditioning is obtained using orthogonal bases (when available, cf. [29] about multivariate orthogonal polynomials), and that Algorithm 2 can improve substantially bases that are not "too bad", depending on the level of conditioning of the initial Vandermonde matrix.

Concerning the initial set of points, in the univariate case Theorem 1 gives a clear guideline, and it is worth recalling that the (continuum) Fekete points are known to be asymptotically nearly equally spaced with respect to the arccos metric. In multivariate instances, much less is known about the distribution of Fekete points. Some results have been recently obtained on the spacing of Fekete points in important standard geometries, like the sphere, the ball and the simplex, which can be

**Table 9**
Absolute quadrature and interpolation errors for the Runge function at the approximate Fekete points extracted after 2 refinement iterations (comparison with the true Fekete points).

| Points | | $n = 10$ | $n = 20$ | $n = 30$ | $n = 40$ | $n = 50$ | $n = 60$ |
|---|---|---|---|---|---|---|---|
| Approx Fek | | | | | | | |
| Basis mon | Quadr | 1.2E−02 | 1.1E−04 | 1.9E−06 | 4.7E−08 | 5.3E−07 | 6.8E−07 |
| | Interp | 1.7E+00 | 1.3E−01 | 1.1E−02 | 9.3E−04 | 3.2E−04 | 2.8E−04 |
| Basis Leg | Quadr | 1.2E−02 | 1.1E−04 | 1.9E−06 | 2.3E−09 | 8.2E−10 | 2.5E−10 |
| | Interp | 1.7E+00 | 1.3E−01 | 1.1E−02 | 9.3E−04 | 7.8E−05 | 6.5E−06 |
| Basis Cheb | Quadr | 1.2E−02 | 1.1E−04 | 1.9E−06 | 2.3E−09 | 8.2E−10 | 2.5E−10 |
| | Interp | 1.7E+00 | 1.3E−01 | 1.1E−02 | 9.3E−04 | 7.8E−05 | 6.5E−06 |
| True Fek | Quadr | 8.9E−03 | 6.2E−05 | 4.4E−07 | 3.1E−09 | 2.2E−11 | 1.6E−13 |
| | Interp | 1.6E+00 | 1.3E−01 | 1.1E−02 | 9.5E−04 | 7.8E−05 | 6.7E−06 |

summarized as

$$\frac{c_1}{n} \le \min\{\mathrm{dist}(a,b),\, b \in \mathcal{F}_n,\, b \ne a\} \le \frac{c_2}{n}, \quad \forall a \in \mathcal{F}_n, \tag{15}$$

where $\mathcal{F}_n$ is a set of Fekete points for the domain, $c_1, c_2$ are positive constants and "dist" is the Dubiner (sphere, ball) or the Baran (simplex) metric (which are generalizations of the arccos metric); see [18,30,31] and the references therein. The lower bound holds for the former with $c_1 = \pi/2$ in any compact set, by a general result of Dubiner [15,31].

These results give further support to the conjecture (Bos [32], see also [33]) that "near-optimal" multivariate interpolation points in general compact domains, and in particular Fekete points, are asymptotically nearly equally spaced with respect to the Dubiner metric. We recall that the Dubiner metric on a compact subset $\Omega \subset \mathbb{R}^d$ has the following definition

$$\mathrm{dist}_{\mathcal{D}}(a,b) := \sup\left\{ \frac{|\arccos(p(b)) - \arccos(p(a))|}{\deg(p)},\ \|p\|_\Omega \le 1, \deg(p) \ge 1 \right\}, \tag{16}$$

for every pair of points $a, b \in \Omega$, cf. [15].

Such a distance is known in explicit form only for very few domains, namely for the square

$$\mathrm{dist}_{\mathcal{D}}(a,b) = \max\{|\arccos(a_1) - \arccos(b_1)|, |\arccos(a_2) - \arccos(b_2)|\}, \tag{17}$$

$a = (a_1, a_2), b = (b_1, b_2) \in \Omega = [-1,1]^2$, for the sphere where it turns out to be simply the geodesic distance, and for the disk where it is obtained by projection on a corresponding hemisphere

$$\mathrm{dist}_{\mathcal{D}}(a,b) = \left| \arccos\left( a_1 a_2 + b_1 b_2 + \sqrt{1 - a_1^2 - b_1^2}\sqrt{1 - a_2^2 - b_2^2} \right) \right|, \tag{18}$$

$a, b \in \Omega = \{z = (z_1, z_2) : z_1^2 + z_2^2 \le 1\}$, with natural generalizations in higher dimension to hypercubes, hyperspheres and balls (cf. [18,30,31]).

### 4.1. Approximate Fekete points in the square

Consider now, as a guideline, the case of the square $\Omega = [-1,1]^2$, with $\mathrm{d}\mu = \mathrm{d}x$ for the quadrature problem. It is worth recalling that Fekete points are known in this case only for tensor-product polynomial spaces [34], whereas here we are interested in total degree polynomial spaces.

We can proceed by the following approach, which is still of heuristic nature. Assume that at degree $n$ there are (at least two) Fekete points in a euclidean neighborhood of the boundary with radius $O(1/n)$, a fact that even though not rigorously proved till now, is verified numerically (indeed, Fekete points cluster at the boundary). Then from the spacing inequality (15), which is only conjectured for the square, and the formula of the Dubiner metric (17), it is not difficult to see that there are Fekete points $a \in \mathcal{F}_n$ such that

$$\frac{k_1}{n^2} \le \min\left\{ \max\left( |a_1 - b_1|, |a_2 - b_2| \right),\ b \in \mathcal{F}_n,\ b \ne a \right\} \le \frac{k_2}{n^2}, \tag{19}$$

where $k_1, k_2$ are positive constants. This suggests that, if we compute approximate Fekete points by Algorithms 1 and 2 starting from a uniform grid of points, the spacing of the grid should be $\mathcal{O}(1/n^2)$, i.e. we should use a $\mathcal{O}(n^2 \times n^2)$ grid which has $M = \mathcal{O}(n^4)$ points.

Now, since the dimension of the polynomial space is $N = \dim(\Pi_n^2(\Omega)) = (n+1)(n+2)/2 \sim n^2/2$, and the computational complexity of the QR factorizations is $\mathcal{O}(MN^2)$ flops (cf. [4]), our algorithm for the computation of approximate Fekete points has a cost of $\mathcal{O}(n^8)$ flops and $\mathcal{O}(n^6)$ storage at degree $n$, which soon becomes a very heavy computational load even using optimized routines of numerical linear algebra.

An alternative strategy could be that of starting from points that already have the right spacing as in (15), for example from a grid of $(n+1) \times (n+2)$ Chebyshev–Lobatto points, that are exactly equally spaced (with a $\mathcal{O}(1/n)$ spacing) in the

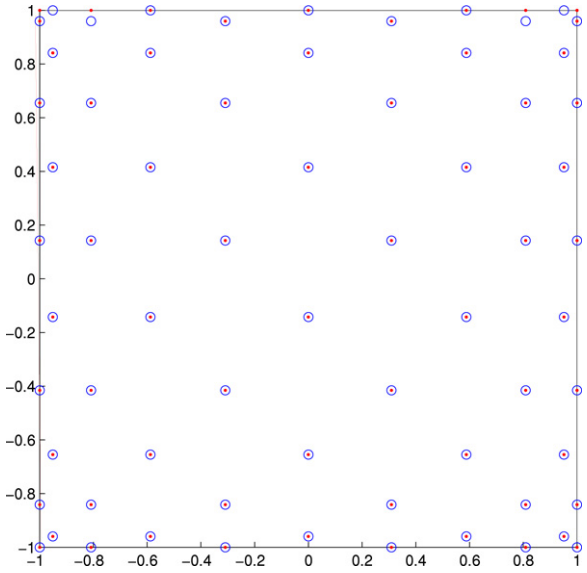


**Fig. 1.** $N = 66$ approximate Fekete points (dots) and Padua points (bullets) from a $11 \times 12$ Chebyshev–Lobatto grid at degree $n = 10$ (they differ only by 4 points close to the top vertices).

Dubiner metric. In such a way we deal with $M = \mathcal{O}(n^2)$ instead of $\mathcal{O}(n^4)$ points, and the computational cost is pulled down by a factor $n^2$ to $\mathcal{O}(n^6)$ flops and $\mathcal{O}(n^4)$ storage. The qualitative idea behind this approach is that Algorithm 1 will then select half of these points as approximate Fekete points, possibly maintaining the correct spacing.

It is worth to stress that both the qualitative strategies sketched above to generate the starting discretization via the conjectured spacing in the Dubiner metric, have indeed also a foundation in the theory of "admissible meshes" for polynomial approximation developed very recently by Calvi and Levenberg in [35]. Both the strategies, in fact, produce (weakly) admissible meshes (at least qualitatively), as tensor products of one-dimensional (weakly) admissible meshes. In [35] it is proved that maximizing the Vandermonde volume on such kind of meshes, gives approximate Fekete points that are (asymptotically) nearly as good as continuum Fekete points. An important fact is that the theory of [35] is applicable to many other multivariate compact domains, e.g. to domains that admit a Markov polynomial inequality, as well as to finite unions and products of such domains.

In Tables 10–12 we show at a sequence of low degrees the cubature and interpolation parameters of approximate Fekete points extracted by Algorithm 2 from a $(2n^2 + 1) \times (2n^2 + 1)$ uniform grid, and of those extracted from $(n + 1) \times (n + 2)$ Chebyshev–Lobatto and Gauss–Lobatto grids. Observe that following [35] the former is sufficiently dense to be an admissible mesh (see the proof of Thm.5 there), and the latter are tensor products of one-dimensional weakly admissible meshes. On the other hand, the latter are also (nearly for Gauss–Lobatto) equally spaced in the Dubiner metric (17). We use the product Chebyshev orthogonal basis (cf. [29]), suitably ordered, to construct the Vandermonde matrix. All the tests are done by an Intel-Centrino Duo processor 1.38 GHz with 1 Gb RAM, using Matlab 6.5 under Windows XP. To avoid possible confusion with the examples of the previous sections, we stress that here we extract approximate Fekete points with a fixed polynomial basis from different grids, whereas in the univariate examples we worked with different bases on a fixed grid.

In order to have a meaningful comparison, we report also the absolute value of the Vandermonde determinant (in the product Chebyshev basis) and the Lebesgue constant of the so-called "Padua points", and of a family of "Padua-like" points obtained from the Gauss–Lobatto tensorial grid (PdGL). We recall that the Padua points, recently studied in [33,36–39], are the first known optimal family for total degree bivariate polynomial interpolation, with a Lebesgue constant growing like $\mathcal{O}(\log^2 n)$. At a fixed degree $n$, they are the union of the two subgrids of a $(n+1) \times (n+2)$ Chebyshev–Lobatto grid, obtained by alternating odd and even indexes in one direction, see Fig. 1 (indeed, there are four families of such points, one obtainable from the other by suitable rotations of the square). Their optimality is intimately related to the fact that they lie on a peculiar algebraic "generating" curve, see [36]. The "Padua-like" Gauss–Lobatto points, obtained in the same way as union of two subgrids of a $(n + 1) \times (n + 2)$ Gauss–Lobatto grid (see Fig. 2), are considered here for the first time.

We can see that the best approximate Fekete points seem those extracted from the Chebyshev–Lobatto tensorial grid in Table 12. The Lebesgue constants are estimated numerically on a $100 \times 100$ uniform control grid (a strange peak of the Lebesgue constant arises at degree $n = 16$ after the iterative refinement, which however is present also in Table 11). The quadrature weights are not all positive, but their $\ell^1$-norm remains close to 4 (the square area), that is the negative weights are few and small. In Tables 10 and 11 we see that also extraction from the other grids gives good results, but with the (much more dense) uniform grid we begin to have memory allocation problems already at relatively low degree ($n = 16$ for iter $> 0$ and $n = 20$ already for iter $= 0$: see the machine features quoted above).

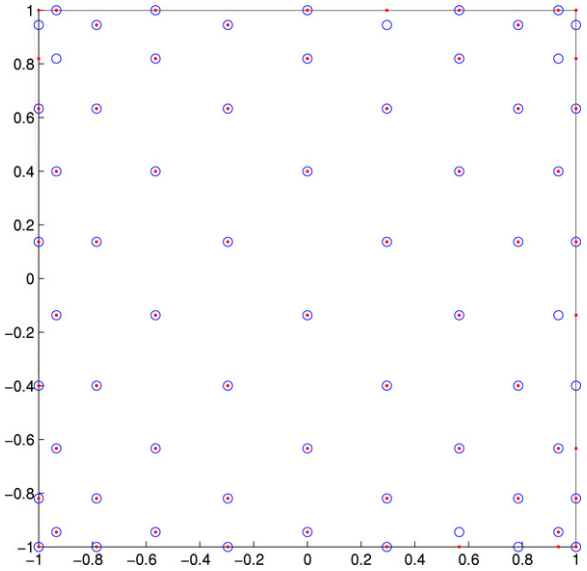

**Fig. 2.** $N = 66$ approximate Fekete points (dots) and Padua-like points (bullets) from a $11 \times 12$ Gauss–Lobatto grid at degree $n = 10$ (they differ only by 9 points close to the boundary).

**Table 10**
Quality parameters of $N = ((n+1) \times (n+2))/2$ approximate Fekete points extracted from a $(2n^2+1) \times (2n^2+1)$ uniform grid in $\Omega = [-1, 1]^2$ with iterative refinement (product Chebyshev basis); $*$ = "out of memory".

| | iter | $n = 4$ ($N = 15$) | $n = 8$ ($N = 45$) | $n = 12$ ($N = 91$) | $n = 16$ ($N = 153$) | $n = 20$ ($N = 231$) |
|---|---|---|---|---|---|---|
| $\mathrm{cond}(V^t)$ | 0 | 2.1E+00 | 2.1E+00 | 2.1E+00 | 2.1E+00 | $*$ |
| | 1 | 1.0E+00 | 1.0E+00 | 1.0E+00 | $*$ | $*$ |
| | 2 | 1.0E+00 | 1.0E+00 | 1.0E+00 | $*$ | $*$ |
| $\lvert\det(V_*)\rvert$ | 0 | 3.0E+06 | 1.3E+27 | 4.9E+65 | 2.2E+125 | $*$ |
| | 1 | 3.0E+06 | 7.8E+26 | 3.6E+66 | $*$ | $*$ |
| | 2 | 3.0E+06 | 7.8E+26 | 3.6E+66 | $*$ | $*$ |
| Padua pts $\lvert\det\rvert$ | | 2.0E+06 | 1.6E+27 | 6.7E+66 | 5.0E+127 | 4.2E+211 |
| $\lVert\mathrm{res}(w)\rVert_2$ | 0 | 2.2E−15 | 3.9E−15 | 5.9E−15 | 6.0E−15 | $*$ |
| | 1 | 1.7E−16 | 5.7E−17 | 2.6E−17 | $*$ | $*$ |
| | 2 | 9.5E−17 | 5.0E−17 | 2.4E−17 | $*$ | $*$ |
| $\sum \lvert w_{i_k}\rvert$ | 0 | 5.05 | 4.32 | 5.31 | 4.11 | $*$ |
| | 1 | 6.34 | 5.45 | 4.24 | $*$ | $*$ |
| | 2 | 6.34 | 4.95 | 4.20 | $*$ | $*$ |
| $\Lambda_n$ | 0 | 5.27 | 10.6 | 22.6 | 28.1 | $*$ |
| | 1 | 5.59 | 11.4 | 12.1 | $*$ | $*$ |
| | 2 | 5.59 | 11.4 | 12.1 | $*$ | $*$ |
| Padua pts $\Lambda_n$ | | 4.41 | 6.21 | 7.45 | 8.41 | 9.20 |

The use of Chebyshev–Lobatto tensorial grids allows us to work without problems at much higher degrees, as we can see in Tables 13 and 14. We report the Lebesgue constants of the approximate Fekete points (after one refinement iteration) at degrees 10, 20, . . . , 60, and the corresponding quadrature and interpolation errors on the bivariate Runge test function, compared to those of the Padua and PdGL points. Notice the very good behavior of the quadrature formulas, which turn out to be more accurate than tensor-product Gaussian formulas and even of the few known minimal formulas, a phenomenon already discussed in [40] (see the error curves in Section 3.3 there).

### *4.2. Conclusions and perspectives*

We have implemented (in Matlab) a method for computing approximate Fekete points, which is essentially based on a greedy algorithm for discrete maximization of Vandermonde determinants and uses only optimized tools of numerical linear algebra (QR-type factorizations of Vandermonde matrices on suitable discretization grids). The choice of the polynomial basis seems to play an important computational role, but it can be corrected and stabilized by discrete orthogonalization. See Algorithms 1 and 2 in Sections 1 and 3.

**Table 11**
Quality parameters of $N = ((n+1)\times(n+2))/2$ approximate Fekete points extracted from a $(n+1)\times(n+2)$ Gauss–Lobatto grid in $\Omega = [-1,1]^2$ with iterative refinement (product Chebyshev basis); PdGL = Padua-like Gauss–Lobatto points.

| | iter | $n=4$ ($N=15$) | $n=8$ ($N=45$) | $n=12$ ($N=91$) | $n=16$ ($N=153$) | $n=20$ ($N=231$) |
|---|---|---|---|---|---|---|
| $\mathrm{cond}(V^t)$ | 0 | 2.6E+00 | 2.6E+00 | 2.5E+00 | 2.4E+00 | 2.3E+00 |
| | 1 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.00E+00 |
| | 2 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.00E+00 |
| $\lvert\det(V_*)\rvert$ | 0 | 2.0E+06 | 5.0E+26 | 5.4E+65 | 7.7E+124 | 1.0E+207 |
| | 1 | 1.8E+06 | 1.3E+27 | 1.7E+65 | 6.6E+123 | 1.9E+209 |
| | 2 | 1.8E+06 | 1.3E+27 | 1.7E+65 | 6.6E+123 | 1.9E+209 |
| PdGL pts $\lvert\det\rvert$ | | 1.9E+06 | 1.3E+27 | 4.2E+66 | 2.2E+127 | 1.2E+211 |
| Padua pts $\lvert\det\rvert$ | | 2.0E+06 | 1.6E+27 | 6.7E+66 | 5.0E+127 | 4.2E+211 |
| $\lVert res(w)\rVert_2$ | 0 | 2.2E−15 | 4.0E−15 | 5.4E−15 | 5.1E−15 | 7.3E−15 |
| | 1 | 6.7E−16 | 5.8E−16 | 6.2E−16 | 4.2E−16 | 4.1E−16 |
| | 2 | 6.7E−16 | 5.8E−16 | 6.2E−16 | 4.2E−16 | 4.1E−16 |
| $\sum \lvert w_{i_k}\rvert$ | 0 | 5.11 | 4.56 | 4.43 | 4.56 | 4.40 |
| | 1 | 6.40 | 4.19 | 4.63 | 4.75 | 4.11 |
| | 2 | 6.40 | 4.19 | 4.63 | 4.75 | 4.11 |
| $\Lambda_n$ | 0 | 6.53 | 12.5 | 15.5 | 26.0 | 34.9 |
| | 1 | 5.83 | 7.99 | 17.0 | 39.3 | 22.7 |
| | 2 | 5.83 | 7.99 | 17.0 | 39.3 | 22.7 |
| PdGL pts $\Lambda_n$ | | 5.20 | 8.53 | 11.3 | 13.7 | 15.8 |
| Padua pts $\Lambda_n$ | | 4.41 | 6.21 | 7.45 | 8.41 | 9.20 |

**Table 12**
Quality parameters of $N = ((n+1)\times(n+2))/2$ approximate Fekete points extracted from a $(n+1)\times(n+2)$ Chebyshev–Lobatto grid in $\Omega = [-1,1]^2$ with iterative refinement (product Chebyshev basis).

| | iter | $n=4$ ($N=15$) | $n=8$ ($N=45$) | $n=12$ ($N=91$) | $n=16$ ($N=153$) | $n=20$ ($N=231$) |
|---|---|---|---|---|---|---|
| $\mathrm{cond}(V^t)$ | 0 | 2.6E+00 | 2.6E+00 | 2.5E+00 | 2.4E+00 | 2.3E+00 |
| | 1 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 |
| | 2 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 | 1.0E+00 |
| $\lvert\det(V_*)\rvert$ | 0 | 1.4E+06 | 6.4E+25 | 3.1E+65 | 3.1E+124 | 6.2E+206 |
| | 1 | 1.4E+06 | 9.9E+26 | 4.5E+66 | 1.7E+125 | 2.9E+211 |
| | 2 | 1.4E+06 | 9.9E+26 | 4.5E+66 | 1.7E+125 | 2.9E+211 |
| Padua pts $\lvert\det\rvert$ | | 2.0E+06 | 1.6E+27 | 6.7E+66 | 5.0E+127 | 4.2E+211 |
| $\lVert \mathrm{res}(w)\rVert_2$ | 0 | 2.3E−15 | 5.4E−15 | 6.1E−15 | 6.0E−15 | 7.5E−15 |
| | 1 | 8.3E−16 | 5.9E−16 | 5.3E−16 | 5.0E−16 | 4.3E−16 |
| | 2 | 8.3E−16 | 5.9E−16 | 5.3E−16 | 5.0E−16 | 4.3E−16 |
| $\sum \lvert w_{i_k}\rvert$ | 0 | 5.25 | 4.85 | 5.14 | 4.93 | 4.54 |
| | 1 | 8.45 | 4.19 | 4.04 | 4.56 | 4.01 |
| | 2 | 8.45 | 4.19 | 4.04 | 4.56 | 4.01 |
| $\Lambda_n$ | 0 | 6.74 | 19.0 | 20.6 | 30.9 | 32.2 |
| | 1 | 7.09 | 8.48 | 9.54 | 20.2 | 11.2 |
| | 2 | 7.09 | 8.48 | 9.54 | 20.2 | 11.2 |
| Padua pts $\Lambda_n$ | | 4.41 | 6.21 | 7.45 | 8.41 | 9.20 |

**Table 13**
Comparison of the Lebesgue constants in $\Omega = [-1,1]^2$: approximate Fekete points (obtained as in Table 12 by 1 refinement iteration), Padua points, Padua-like Gauss–Lobatto points (PdGL).

| Points | $n=10$ ($N=66$) | $n=20$ ($N=231$) | $n=30$ ($N=496$) | $n=40$ ($N=861$) | $n=50$ ($N=1326$) | $n=60$ ($N=1891$) |
|---|---|---|---|---|---|---|
| Approx Fek | 9.01 | 11.2 | 12.9 | 37.9 | 38.2 | 40.6 |
| Padua | 6.88 | 9.20 | 10.7 | 11.9 | 12.9 | 13.7 |
| PdGL | 10.0 | 15.8 | 20.5 | 24.6 | 28.2 | 31.5 |

The numerical tests for the interval (Sections 2 and 3) and for the square (Section 4.1) have shown that such approximate Fekete points have very good features for algebraic quadrature and interpolation. Their Lebesgue constants grow slowly, and the associated quadrature weights, even though not all positive, show an $\ell^1$-norm which remains close to the domain measure. The interval and the square are the only domains where optimal interpolation points are theoretically known, and indeed we have used such points for the purpose of comparison (in particular, the recently discovered "Padua points" for the square, see [33,36,37]).

**Table 14**
Absolute quadrature and interpolation errors for the bivariate version of the Runge function: approximate Fekete points (obtained as in Table 12 by 1 refinement iteration), Padua points, Padua-like Gauss–Lobatto points (PdGL); the integral up to machine precision is 0.597388947274307.

| Points | $n = 10$ ($N = 66$) | $n = 20$ ($N = 231$) | $n = 30$ ($N = 496$) | $n = 40$ ($N = 861$) | $n = 50$ ($N = 1326$) | $n = 60$ ($N = 1891$) |
|---|---|---|---|---|---|---|
| Approx Fek | | | | | | |
| Quadr | 2.0E−03 | 6.0E−05 | 2.3E−06 | 4.0E−07 | 1.3E−08 | 2.1E−09 |
| Interp | 3.9E+00 | 6.2E−01 | 9.9E−02 | 1.6E−02 | 1.4E−03 | 2.4E−04 |
| Padua | | | | | | |
| Quadr | 5.2E−04 | 1.3E−05 | 2.1E−07 | 1.3E−08 | 8.0E−10 | 5.9E−11 |
| Interp | 3.9E+00 | 6.2E−01 | 9.9E−02 | 1.6E−02 | 1.4E−03 | 2.4E−04 |
| PdGL | | | | | | |
| Quadr | 4.1E−04 | 1.3E−05 | 3.6E−07 | 2.4E−08 | 1.8E−09 | 1.6E−10 |
| Interp | 3.8E+00 | 5.9E−01 | 9.5E−02 | 1.6E−02 | 1.5E−03 | 2.5E−04 |

In the examples we have used a connection with the theory of metrics associated to polynomial inequalities and with the very recent theory of admissible meshes for polynomial approximation, see [18,31,35]. This connection, together with the relatively low computational complexity of the discrete method with respect to very costly maximizations in the continuum (cf. e.g. [19]), seems to open the way towards computation of good points for global high-degree interpolation and quadrature in many other standard and nonstandard multivariate domains. On the other hand, also local polynomial methods based on piecewise interpolation over standard and nonstandard partitions of the domains could benefit from this new approach.

## References


[1] A. Sommariva, M. Vianello, Product Gauss cubature over polygons based on Green's integration formula, BIT 47 (2007) 441–453.
[2] A. Sommariva, M. Vianello, Gauss–Green cubature and moment computation over arbitrary geometries, preprint, 2008 (available online at: http://www.math.unipd.it/~marcov/publications.html).
[3] A. Björk, Numerical Methods for Least Squares Problems, SIAM, 1996.
[4] G.H. Golub, C.F. Van loan, Matrix Computations, third ed., Johns Hopkins University Press, 1996.
[5] The Mathworks, MATLAB documentation set 2007 version (available online at: http://www.mathworks.com).
[6] W. Gautschi, Moments in quadrature problems, Comput. Math. Appl. 33 (1997) 105–118.
[7] P.A. Businger, G.H. Golub, Linear least-squares solutions by householder transformations, Numer. Math. 7 (1965) 269–276.
[8] LAPACK Users' Guide, SIAM, 1999, (available online at http://www.netlib.org/lapack).
[9] L. Bos, N. Levenberg, On the calculation of approximate Fekete points: the univariate case, Electron. Trans. Numer. Anal. 30 (2008) 377–397.
[10] V.I. Krylov, Approximate Calculation of Integrals, Macmillan, 1962.
[11] T.J. Rivlin, An Introduction to the Approximation of Functions, Dover, 1969.
[12] J.C. Mason, D.C. Handscomb, Chebyshev Polynomials, Chapman & Hall, 2003.
[13] A. Civril, M. Magdon-Ismail, Finding maximum volume sub-matrices of a matrix, Technical Report 07–08, Dept. of Comp. Sci., RPI, 2007 (available online at: www.cs.rpi.edu/research/pdf/07-08.pdf).
[14] G. Quintana-Ortí, X. Sun, C.H. Bischof, A BLAS-3 version of the QR factorization with column pivoting, SIAM J. Sci. Comput. 19 (1998) 1486–1494.
[15] M. Dubiner, The theory of multidimensional polynomial approximation, J. Anal. Math. 67 (1995) 39–116.
[16] M. Reimer, Multivariate Polynomial Approximation, Birkhäuser, 2003.
[17] B. Sündermann, Lebesgue constants in Lagrangian interpolation at the Fekete points, Mitt. Math. Ges. Hamburg 11 (1983) 204–211.
[18] L. Bos, N. Levenberg, S. Waldron, On the spacing of Fekete points for a sphere, ball or simplex, Indag. Math. 19 (2008) 163–176.
[19] M.A. Taylor, B.A. Wingate, R.E. Vincent, An algorithm for computing Fekete points in the triangle, SIAM J. Numer. Anal. 38 (2000) 1707–1720.
[20] R. Pasquetti, F. Rapetti, Spectral element methods on unstructured meshes: Comparisons and recent advances, J. Sci. Comput. 27 (2006) 377–387.
[21] J. Burkardt, Fekete: High order interpolation and quadrature in triangles, Matlab/Fortran90/C++ libraries (available online at: http://people.scs.fsu.edu/~burkardt/m_src/fekete/fekete.html).
[22] I.H. Sloan, R.S. Womersley, Extremal systems of points and numerical integration on the sphere, Adv. Comput. Math. 21 (2004) 107–125.
[23] R. Cools, An encyclopedia of cubature formulas, J. Complexity 19 (2003) 445–453.
[24] M.A. Taylor, B.A. Wingate, L. Bos, A cardinal function algorithm for computing multivariate quadrature points, SIAM J. Numer. Anal. 45 (2007) 193–205.
[25] T. Kövari, C. Pommerenke, On the distribution of Fekete points, Mathematika 15 (1968) 70–75.
[26] D. Fasino, G. Inglese, On the spectral condition of rectangular Vandermonde matrices, Calcolo 29 (1992) 291–300.
[27] W. Gautschi, How (un)stable are Vandermonde systems?, in: R. Wong (Ed.), Asymptotic and Computational Analysis, in: Lecture Notes in Pure and Applied Mathematics, vol. 124, Marcel Dekker, 1990, pp. 193–210.
[28] 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.
[29] C.F. Dunkl, Y. Xu, Orthogonal Polynomials of Several Variables, in: Encyclopedia of Mathematics and its Applications, vol. 81, Cambridge University Press, 2001.
[30] L. Bos, N. Levenberg, S. Waldron, Metrics associated to multivariate polynomial inequalities, in: M. Neamtu, E.B. Saff (Eds.), Advances in Constructive Approximation, Nashboro Press, Nashville, 2004, pp. 133–147.
[31] L. Bos, N. Levenberg, S. Waldron, Pseudometrics, distances and multivariate polynomial inequalities, J. Approx. Theory 153 (2008) 80–96.
[32] L. Bos, Multivariate interpolation and polynomial inequalities, Ph.D. course held at the University of Padua, 2001, unpublished.
[33] M. Caliari, S. De Marchi, M. Vianello, Bivariate polynomial interpolation on the square at new nodal sets, Appl. Math. Comput. 161 (2005) 261–274.
[34] L. Bos, M.A. Taylor, B.A. Wingate, Tensor product Gauss–Lobatto points are Fekete points for the cube, Math. Comp. 70 (2001) 1543–1547.
[35] J.P. Calvi, N. Levenberg, Uniform approximation by discrete least squares polynomials, J. Approx. Theory 152 (2008) 82–100.
[36] L. Bos, M. Caliari, S. De Marchi, M. Vianello, Y. Xu, Bivariate Lagrange interpolation at the Padua points: The generating curve approach, J. Approx. Theory 143 (2006) 15–25.

[37] L. Bos, S. De Marchi, M. Vianello, Y. Xu, Bivariate Lagrange interpolation at the Padua points: The ideal theory approach, Numer. Math. 108 (2007) 43–57.
[38] M. Caliari, S. De Marchi, M. Vianello, Bivariate Lagrange interpolation at the Padua points: Computational aspects, J. Comput. Appl. Math. 221 (2008) 284–292.
[39] M. Caliari, S. De Marchi, M. Vianello, Padua2D: Bivariate Lagrange interpolation at Padua points on bivariate domains, ACM Trans. Math. Software 35-3 (2008).
[40] A. Sommariva, M. Vianello, R. Zanovello, Nontensorial Clenshaw–Curtis cubature, Numer. Algorithms 49 (2008) 409–427.