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 Ω), say

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

and a sufficiently large and dense discretization of Ω

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

we can construct the rectangular Vandermonde matrix

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

A quadrature formula of algebraic degree of exactness n for a given measure μ on Ω 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*).

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

where $\mathrm{ind} = (i_1, \ldots, i_N)$, 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 Λ_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