A method to compute recurrence relation coefficients for bivariate orthogonal polynomials by unitary matrix transformations

Marc Van Barel   *Andrey Chesnokov*

Report TW 554, December 2009

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

A method to compute recurrence relation coefficients for bivariate orthogonal polynomials by unitary matrix transformations

Marc Van Barel *Andrey Chesnokov*

Report TW 554, December 2009

Department of Computer Science, K.U.Leuven

Abstract

We present an algorithm computing recurrence relation coefficients for bivariate polynomials, orthonormal with respect to a discrete inner product. These polynomials make it possible to give the solution of a discrete least squares approximation problem. To compute these polynomials, we pose the inverse eigenvalue problem and solve it efficiently and in a stable way, using a sequence of Givens rotations. We also show how to generalize the algorithm for the case of polynomials in more variables. Several numerical experiments show the validity of the approach.

Keywords : orthogonal polynomials, recurrence relations, unitary transformations, Givens rotations, discrete least squares problem, bivariate interpolation and approximation, Padua points.
MSC : Primary : 42C05, Secondary : 65F18, 65F25.

Noname manuscript No.
(will be inserted by the editor)

A method to compute recurrence relation coefficients for bivariate orthogonal polynomials by unitary matrix transformations

Marc Van Barel · **Andrey Chesnokov**

the date of receipt and acceptance should be inserted later

Abstract We present an algorithm computing recurrence relation coefficients for bivariate polynomials, orthonormal with respect to a discrete inner product. These polynomials make it possible to give the solution of a discrete least squares approximation problem. To compute these polynomials, we pose the inverse eigenvalue problem and solve it efficiently and in a stable way, using a sequence of Givens rotations. We also show how to generalize the algorithm for the case of polynomials in more variables. Several numerical experiments show the validity of the approach.

Keywords orthogonal polynomials, recurrence relations, unitary transformations, Givens rotations, discrete least squares problem, bivariate interpolation and approximation, Padua points

1 Introduction

Let $\{\zeta_k\}_{k=1}^m$ be a set of nodes – pairs of complex numbers and $\{w_k^2\}_{k=0}^m$ a set of positive weights (let us assume that $w_k > 0$). We solve the problem of finding the least squares multivariate polynomial approximant in the space $\mathscr{P}$ with positive semidefinite inner product

$$\langle p,q\rangle = \sum_{i=0}^{m} w_i^2 \overline{p(\zeta_i)} q(\zeta_i). \tag{1}$$

This is a positive definite inner product for the space of vectors $(p(\zeta_0),\ldots,p(\zeta_m))$ representing the function values at the given nodes.

The work is partially supported by the Research Council K.U.Leuven, project OT/05/40 (Large rank structured matrix computations), CoE EF/05/006 Optimization in Engineering (OPTEC), and by the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with the authors.

Marc Van Barel · Andrey Chesnokov
Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium. E-mail: marc.vanbarel@cs.kuleuven.be, andrey.chesnokov@cs.kuleuven.be

The polynomial $p \in \mathscr{P}$ that minimizes

$$\| f - p\|$$

can be found as follows. Find a basis $\{\phi_1, \ldots, \phi_n\}$ for $\mathscr{P}$ which is orthonormal with respect to $\langle \cdot, \cdot \rangle$. The solution p is the generalized Fourier expansion of f with respect to this basis, truncated after some term. An algorithm that solves this problem will compute implicitly or explicitly the orthonormal basis and the Fourier coefficients. For the univariate case the nodes are just complex numbers. Special choices of ζ_i (like when all the ζ_i are on the real line or all the ζ_i on the complex unit circle) lead to a "short recurrence" for the orthogonal polynomials and thus reduce the complexity of such an algorithm. However, in the multivariate case the number of nonzero terms in the recurrence relation is growing as the degree of the polynomial grows. More details on recurrence relations between univariate polynomials orthogonal on the real line are presented in the book of Gautschi [11]. Plenty of the applications where different orthonormal polynomials are a valuable and fruitful tool are discussed in the book of Golub and Meurant [12].

To solve the above-stated least squares problem we compute recurrence relation coefficients of the orthonormal basis polynomials and in parallel the Fourier coefficients of the expansion of f. The recurrence relation coefficients are coming from two coupled inverse eigenvalue problems, which we solve by means of a sequence of unitary similarity transformations.

For a survey of methods on inverse eigenvalue problems, we refer to Chu and Golub [8]. When all ζ_i are on the real line, the discrete least squares interpretations of these methods are presented by Reichel [13] and by Elhay, Golub and Kautsky [9]. These methods efficiently exploit the tridiagonal structure of the matrix representing the recurrence relation and construct the optimal polynomial fitting in a least squares sense, given the function values in these real points ζ_i . Based on the inverse unitary *QR* algorithm for computing unitary Hessenberg matrices [1], Reichel, Ammar and Gragg [14] solve the approximation problem when the given function values are taken in points ζ_i on the unit circle. Their algorithm is based on computational aspects associated with the family of polynomials orthogonal with respect to an inner product on the unit circle. Such polynomials are known as Szegő polynomials. Fassbender [10] presents an approximation algorithm based on an inverse unitary Hessenberg eigenvalue problem and shows that it is equivalent to computing Szegő polynomials. More properties of the inverse unitary Hessenberg eigenvalue problem are studied by Ammar and He [2].

A generalization of these ideas to vector orthogonal polynomials and to the least squares problems of a more general nature is presented by Bultheel and Van Barel in [5,15]. They developed an updating procedure to compute a sequence of orthonormal polynomial vectors with respect to that inner product where the points ζ_i could lie anywhere in the complex plane. Again, if the inner products are prescribed in points on the real axis or on the unit circle, they present variants of the algorithm which are an order of magnitude more efficient.

The method presented in this paper generalizes these ideas to the bivariate case. The points where the inner products are prescribed may constitute any pairs of complex points. However, when we take pairs of real points, the generalized Hessenberg matrices representing the recurrence relations become symmetric. This does not have that significant effect on the complexity like it had in the univariate case, because the lower bandwidth of these matrices is growing together with their size.

For our numerical experiments we use Padua points as the set of nodes for the discrete inner product. These points were introduced for the first time by Caliari, De Marchi and Vianello in [7]. Such points are an example of optimal points with real coordinates for total degree polynomial interpolation in two variables, with a Lebesgue constant increasing like log squared of the degree, see [3,4].

The paper is organized as follows. In Section 2 we state the problem of polynomial least squares approximation and show how the orthogonal polynomials appear naturally in its context. In Section 3 we show how generalized Hessenberg matrices arise in recurrence relations between these orthogonal polynomials and how the original problem of constructing these polynomials is reduced to an inverse eigenvalue problem. In Section 4 we present an algorithm to solve this inverse eigenvalue problem and in Section 5 we give several numerical examples.

2 Polynomial least squares approximation

Discrete least squares approximation by polynomials is a classical problem in numerical analysis where orthogonal polynomials play a central role.

Let us first define the polynomial spaces we are working in. Let $R[\mathbf{x}]$ with $\mathbf{x} = (x_1, x_2)$ be the ring of all polynomials in two variables. Fix a monomial order $\prec$, say the graded lexicographical order, and let $\mathrm{lt}(f)$ denote the leading term of the polynomial $f \in R[\mathbf{x}]$ according to the monomial order. This monomial order induces the order $\prec$ on the polynomials, namely, we say that $p(x,y) \prec q(x,y)$ iff $\mathrm{lt}(p) \prec \mathrm{lt}(q)$. Consider any ordered sequence of terms $t_0 \prec \cdots \prec t_n$ and define by $\mathscr{P}_n$ its linear span. We say that the polynomial $p \in \mathscr{P}_n$ has length k iff all the coefficients at the terms t_i, $k < i \leqslant n$, are zero and $t_k \neq 0$. It is assumed that together with each term $x^i y^j$ all the terms $x^p y^q$, $p \leqslant i$, $q \leqslant j$ are also in $\mathscr{P}_n$ and are preceding $x^i y^j$ wrt chosen monomial order.

Given an inner product $\langle \cdot, \cdot \rangle$ defined on $\mathscr{P}_m \times \mathscr{P}_m$, the polynomial $p \in \mathscr{P}_n$ of length at most $n \leqslant m$, which minimizes the error

$$\|f - p\|, \quad p \in \mathscr{P}_n \tag{2}$$

is given by

$$p = \sum_{k=0}^{n} a_k \phi_k, \quad a_k = \langle f, \phi_k \rangle, \tag{3}$$

where the $\{\phi_k\}_0^n$ form an orthonormal set of polynomials:

$$\phi_k \in \mathscr{P}_k - \mathscr{P}_{k-1}, \quad P_{-1} = \emptyset, \quad \langle \phi_k, \phi_l \rangle = \delta_{kl}.$$

The inner product we consider here is of discrete form (1) where ζ_i are distinct pairs of complex numbers. When $m = n$, the least squares solution is the interpolating polynomial.

Let us illustrate now how the orthogonal polynomials do appear in this context. We start with some multivariate polynomial basis $\{\psi_k\}$, $\psi_k \in \mathscr{P}_k - \mathscr{P}_{k-1}$. Let us set

$$p = \sum_{k=0}^{n} c_k \psi_k, \quad c_k \in \mathbb{C}.$$

Then the least squares problem can be reformulated as finding the weighted least squares solution of the system of linear equations

$$\min_{c_k} \sum_{k=0}^{n} w_i^2 (c_k \psi_k(\zeta_i) - f(\zeta_i))^2, \quad i = 0, \ldots, m,$$

which is the least squares solution of the system

$$\min_{\mathbf{c}_n} \|W(\Psi_n \mathbf{c}_n - \mathbf{f})\|_2,$$

where $W = \operatorname{diag}(w_0, \ldots, w_m)$ and

$$\Psi_n = \begin{pmatrix} \psi_0(\zeta_0) & \ldots & \psi_n(\zeta_0) \\ \vdots & & \vdots \\ \psi_0(\zeta_m) & \ldots & \psi_n(\zeta_m) \end{pmatrix}, \quad \mathbf{c}_n = \begin{pmatrix} b_0 \\ \vdots \\ b_n \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} f(\zeta_0) \\ \vdots \\ f(\zeta_m) \end{pmatrix}. \tag{4}$$

The normal equations for this system are

$$(\Psi_n^H W^H W \Psi_n)\mathbf{c}_n = \Psi_n^H W^H W \mathbf{f}. \tag{5}$$

When the ψ_k are chosen to be the orthonormal polynomials ϕ_k, then $\Psi_n^H W^H W \Psi_n = I_{n+1}$ and the previous system has the solution $\mathbf{c}_n = \Psi_n^H W^H W \mathbf{f}$.

3 Generalized Hessenberg matrices and recurrence relations

From the previous discussion it follows that the central problem is to construct the orthonormal basis $\{\phi_k\}$. We use the abbreviation OP for orthonormal polynomials.

Let us recall the one-variable case. In general, the polynomial $z\phi_{k-1}(z)$ can be expressed as a linear combination of the polynomials $\phi_0, \ldots, \phi_k$, leading to a relation of the form

$$z\phi_{k-1}(z) = \eta_{kk}\phi_k(z) + \cdots + \eta_{0k}\phi_o(z), \quad k = 1, \ldots, m+1.$$

We can express the previous relations as

$$z[\phi_0(z), \ldots, \phi_m(z)] = [\phi_0(z), \ldots, \phi_m(z)]H + e_{m+1}^T \phi_{m+1}(z)\eta_{m+1,m+1}, \tag{6}$$

where H is an upper Hessenberg matrix and $e_{m+1}^T = [0, 0, \ldots, 0, 1]$.

Since we identify the function with the $(m+1)$-vector of its function values in ζ_k, $k = 0, \ldots, m$ (being just complex numbers in the one-dimensional case), our "functional space" is a space of $(m+1)$-vectors, which is $(m+1)$-dimensional. This means that the $(m+2)$-nd orthogonal polynomial will be orthogonal to the whole space, hence it must be zero. Thus, if ϕ_k are these orthogonal polynomials, then $[\phi_{m+1}(\zeta_0),\ldots,\phi_{m+1}(z_m)]^T$ is the zero vector. Hence, $\phi_{m+1}(z) = \Pi_i(z-\zeta_i)$.

Let us set define the matrix Φ_m similarly to Ψ_m (4), replacing polynomials ψ with ϕ. Let us set $\Phi = \Phi_m$ and rewrite relation (6) as

$$Z\Phi = \Phi H,$$

with $Z = \mathrm{diag}(\zeta_0,\ldots,\zeta_m)$.

Multiplying with the diagonal matrix W and using $ZW = WZ$, we are led to

$$H = (W\Phi)^H Z(W\Phi) = Q^H Z Q, \tag{7}$$

which means that the diagonal matrix Z and the Hessenberg matrix H are unitarily similar. The constant polynomial ϕ_0 is normalized when it is equal to η_{00}^{-1} with η_{00} given by

$$Q^H\mathbf{w} = [\eta_{00},0,\ldots,0]^T,$$

where $\mathbf{w} = [w_0,\ldots,w_m]^T$. Since $Q = W\Phi$ and $\|\phi_0\| = 1$, we see that all the entries in $Q^H\mathbf{w}$ are zero by orthogonality, except for the first one, which is $1/\phi_0$.

Thus the problem of constructing a one-variable orthonormal polynomial basis is reduced to the following inverse eigenvalue problem: given the complex points $Z = \mathrm{diag}(\zeta_i)$ and the weights $\mathbf{w} = (w_i)$, find unitary Q and upper Hessenberg H such that

$$Q^H\mathbf{w} = \|\mathbf{w}\|\mathbf{e}_1, \quad Q^H Z Q = H. \tag{8}$$

To generalize this to the bivariate case we must recall that we have a choice between multiplication by x and y to proceed from a current OP $\phi_{k-1}(x,y)$ to one of the following OPs. This choice is predetermined by the ordering of terms chosen in the definition of $\mathscr{P}_n$.

Recurrence relations for the two-variable OPs could be written in a manner similar to (6):

$$\begin{aligned} x[\phi_0(x,y),\phi_1(x,y),\phi_2(x,y),\ldots] &= [\phi_0(x,y),\phi_1(x,y),\phi_2(x,y),\ldots]H_x,\\ y[\phi_0(x,y),\phi_1(x,y),\phi_2(x,y),\ldots] &= [\phi_0(x,y),\phi_1(x,y),\phi_2(x,y),\ldots]H_y. \end{aligned} \tag{9}$$

However, the matrices H_x and H_y are not anymore Hessenberg. They are what we call generalized Hessenberg and their structure becomes clear from the following example.

Consider the graded lexicographic ordering of terms (the numbers in the second table are the order numbers of terms in the same place in the first table):

$$\begin{array}{c|ccccc} & y^4 & & & & \\ & y^3 & xy^3 & & & \\ y & y^2 & xy^2 & x^2y^2 & & \\ & y & xy & x^2y & x^3y & \\ & 1 & x & x^2 & x^3 & x^4 \\ \hline & & & x & & \end{array} \qquad \begin{array}{c|ccccc} & 15 & & & & \\ & 10 & 14 & & & \\ y & 6 & 9 & 13 & & \\ & 3 & 5 & 8 & 12 & \\ & 1 & 2 & 4 & 7 & 11 \\ \hline & & & x & & \end{array}$$

.

Let us write the recurrence relations for the first six polynomials, denoting by boldface the polynomial that is being determined from each equation:

$$
\begin{array}{c}
\boldsymbol{\phi_1} = const \\
x\phi_1 = [\phi_1, \boldsymbol{\phi_2}] \cdot H_x(1:2,1) \\
y\phi_1 = [\phi_1, \phi_2, \boldsymbol{\phi_3}] \cdot H_y(1:3,1) \\
x\phi_2 = [\phi_1, \phi_2, \phi_3, \boldsymbol{\phi_4}] \cdot H_x(1:4,2) \\
y\phi_2 = [\phi_1, \phi_2, \phi_3, \phi_4, \boldsymbol{\phi_5}] \cdot H_y(1:5,2) \\
x\phi_3 = [\phi_1, \phi_2, \phi_3, \phi_4, \boldsymbol{\phi_5}] \cdot H_x(1:5,3) \\
y\phi_3 = [\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \boldsymbol{\phi_6}] \cdot H_y(1:6,3)
\end{array}
\tag{10}
$$

For the case of real ζ_i the matrices H_x and H_y are symmetric and have the following structure ($\times$, $\bowtie$ and $\boxtimes$ all denote (possibly) nonzero elements, $\boxtimes$ is the pivot essential for the previous system of equations):

$$
H_x = \begin{bmatrix}
\times & \bowtie & & & & \\
\boxtimes & \times & \bowtie & \bowtie & & \\
 & \bowtie & \times & \bowtie & \bowtie & \bowtie \\
 & \boxtimes & \bowtie & \times & \bowtie & \bowtie \\
 & & \boxtimes & \bowtie & \times & \bowtie \\
 & & & \bowtie & \bowtie & \times
\end{bmatrix}, \quad
H_y = \begin{bmatrix}
\times & \bowtie & \bowtie & & & \\
\bowtie & \times & \bowtie & \bowtie & \bowtie & \\
\boxtimes & \bowtie & \times & \bowtie & \bowtie & \bowtie \\
 & \bowtie & \bowtie & \times & \bowtie & \bowtie \\
 & \boxtimes & \bowtie & \bowtie & \times & \bowtie \\
 & & \boxtimes & \bowtie & \bowtie & \times
\end{bmatrix}.
\tag{11}
$$

Polynomial $\phi_5(\zeta)$ is the first polynomial that could be reached by multiplication both by x and y. This gives two equations to determine it. The following scheme illustrates one of the possibilities (we call it downside-up scheme). Vertical arrows denote multiplication by y, horizontal arrows denote multiplication by x.

$$
\begin{array}{c|ccccccc}
 & 10 & & & & & & \\
 & \uparrow & & & & & & \\
 & 6 & & 9 & & & & \\
y & \uparrow & & \uparrow & & & & \\
 & 3 & & 5 & & 8 & & \\
 & \uparrow & & \uparrow & & \uparrow & & \\
 & 1 & \rightarrow & 2 & \rightarrow & 4 & \rightarrow & 7 \\
\hline
 & & & x & & & &
\end{array}
\tag{12}
$$

Given the monomial ordering the structure of the matrices H_x and H_y is fixed. If one of the pivot elements $\boxtimes$ becomes zero on some step, this means that on this step an interpolating polynomial is constructed. This polynomial is zero in all used up till this step points ζ_i.

To determine the basis polynomial ϕ_k we may introduce the following inductive scheme. Basis: fix ϕ_0 to be constant. Inductive step: suppose that all ϕ_i, $0 \leqslant i < k$, are computed. For ϕ_k look at the scheme (12) and decide whether to take coefficients from H_x or H_y matrix. Take the first unused column in this matrix and build on its basis the recurrence equation like in (10). Mark the column as used. If there is a recurrence relation for the same polynomial in the second matrix, then mark the first unused column there as used also.

This scheme is easily generalized to polynomials of more than two variables. The only thing that needs to be updated is the monomial order and the scheme (12), determining the recurrence relations.

Let us now derive the inverse eigenvalue problem, similar to (8) in the univariate case.

We may recall again that our inner product is discrete and thus we work with functions as with the $(m+1)$-vector of its values in $\zeta_k = (x_k, y_k)$, $k = 0, \dots, m$. It means that the $(m+1)$-st orthogonal polynomial will be orthogonal to the whole space, hence it must be zero. This makes possible to rewrite the relations (9) as

$$X\Phi = \Phi H_x,$$
$$Y\Phi = \Phi H_y,$$

with

$$\Phi = \Phi_m = \begin{pmatrix} \phi_0(\zeta_0) & \dots & \phi_n(\zeta_0) \\ \vdots & & \vdots \\ \phi_0(\zeta_m) & \dots & \phi_n(\zeta_m) \end{pmatrix}.$$

Using a similar technique as in the beginning of this section, the problem of constructing the bivariate orthonormal polynomial basis is reduced to the following inverse eigenvalue problem: given the complex points $\boldsymbol{\zeta}_i = (x_i, y_i)$, $X = \mathrm{diag}(x_i)$, $Y = \mathrm{diag}(y_i)$, the weights $\mathbf{w} = (w_i)$ and the ordering scheme, find unitary Q and upper generalized Hessenberg matrices H_x and H_y such that

$$Q^H \mathbf{w} = \|\mathbf{w}\|\mathbf{e}_1, \quad Q^H X Q = H_x, \quad Q^H Y Q = H_y. \tag{13}$$

4 Inverse eigenvalue problem and updating algorithm

4.1 General formulation of the algorithm

In this section, we give an algorithm which computes given the initial data (the points $\boldsymbol{\zeta}_i$, the weights $\mathbf{w}_i$ and the ordering scheme $\pi(i)$) the matrices H_x and H_y – the building blocks of the requrrence relation generating the desired orthonormal polynomials.

The algorithm starts with the following matrix:

$$\begin{pmatrix} w_1 & x_1 & & & & y_1 & & & \\ w_2 & & x_2 & & & & y_2 & & \\ \vdots & & & \ddots & & & & \ddots & \\ w_m & & & & x_m & & & & y_m \end{pmatrix} = [\mathbf{w}|X|Y] \in \mathbb{C}^{m\times(2m+1)} \tag{14}$$

and transforms it using unitary similarity transformations into

$$[\|\mathbf{w}\|\mathbf{e}_1|H_x|H_y] = [Q^H\mathbf{w}|Q^H XQ|Q^H YQ] = Q^H[\mathbf{w}|X|Y]\begin{pmatrix} I_n & & \\ & Q & \\ & & Q \end{pmatrix}. \tag{15}$$

The matrix Q is unitary, such that $[Q^H\mathbf{w}|Q^H XQ|Q^H YQ]$ has zeros below the pivot positions $(i, \pi(i))$, $i = 1, 2, \dots, m$. The following algorithm will add, for each i, the point $\boldsymbol{\zeta}_i$ with corresponding weight w_i. Each iteration changes the underlying inner product, adding a new point ζ_i to it.

Algorithm 1: Transformation of $D = [\mathbf{w}|X|Y]$ into $[Q^H\mathbf{w}|Q^H XQ|Q^H YQ]$ having zeros below the pivots, function $\pi(j)$ is a $\mathbf{w}/x/y$ switch

---

begin
- **for** $i = 2 : n$ **do**
  - **for** $j = 1 : i-1$ **do**
    - make element $d_{i,\pi(j)}$ zero by Givens rotation G^H with the pivot element $(j, \pi(j))$: $D = G^H D$
    - $D = D \begin{pmatrix} 1 & & \\ & G & \\ & & G \end{pmatrix}$ (similarity transformation)

    end

  end

end

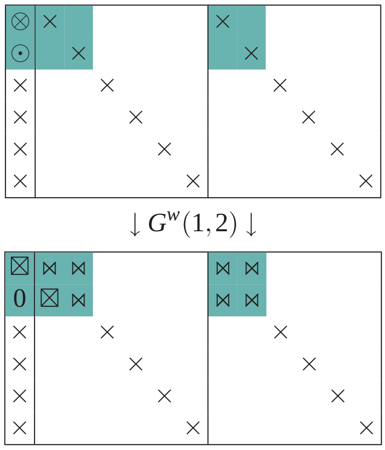
4.2 6×6 example

Let us restrict ourselves to the problem of finding the first 6 OPs using the downside-up ordering scheme (12). Here $\times$ denotes the original elements of the matrix $D = [\mathbf{w}|X|Y]$, $\bowtie$ stands for some possible nonzero elements, $\boxtimes$ marks the positions of pivots essential for the recurrence relations, and at each step $\odot$ is the element to be annihilated by the Givens rotation built on the element $\otimes$. By we mark the part of the matrix D that is actually changing at the corresponding *inner* iteration of Algorithm 1. Elements marked by together with elements marked by represent the part of the matrix D that is being processed during the current *outer* iteration of Algorithm 1.

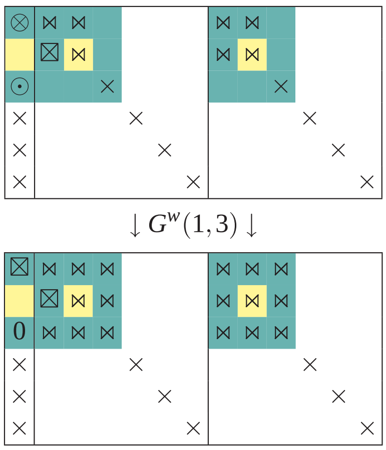
We also specify the transformation matrix at each step. $G^w(i, j)$ means the Givens rotation constructed using the i-th and j-th element of the (transformed) weight vector; $G^X(i, j)$ means the Givens rotation constructed using an element in the X part of the matrix D, placed in i-th row to annihilate the corresponding element placed in j-th row; the column is then $\pi(j)$, and similarly to it we define $G^Y(i, j)$.

We restrict ourselves to pairs of real points $\boldsymbol{\zeta}_i$, resulting in matrices H_x and H_y being symmetric.

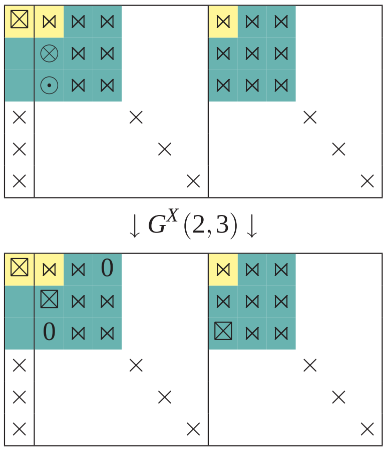
We start with the inner product based on 2 points, so 2×2-submatrices of D are being processed. The ordering scheme tells that the second polynomial comes from the X part, and we mark there its corresponding pivot with $\boxtimes$.



Then we add one more point and start again with the weight vector.

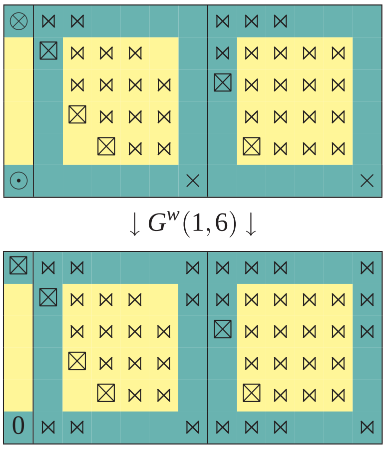


At this moment we have to restore the (generalized) Hessenberg structure in the first column of the X part, using the pivot. Because of symmetry two zeros will appear in the X part. Then we mark the pivot position for the third polynomial, it is in the Y part according to (12).

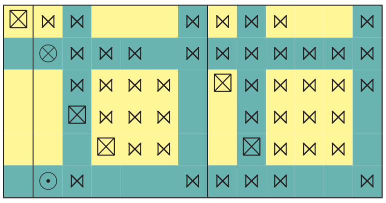


Let us skip the stages of working with 4 and 5 points and proceed directly to the procedure of adding the 6th point. Again we start with the weight vector. Note that there are two pivot positions in the 5th row. This corresponds to the fact that the 5th OP can be derived either by multiplying by x or by y.

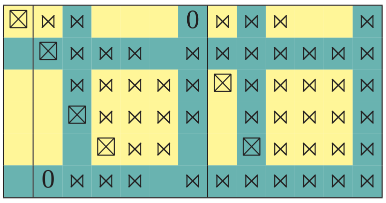
At first, we make a zero element in $\mathbf{w}$-vector:



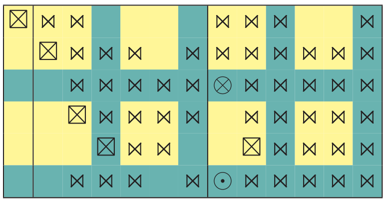
Now the generalized Hessenberg structure is destroyed, so we start chasing the nonzeros in the last row from left to right. The first pivot comes from the X part, second row.



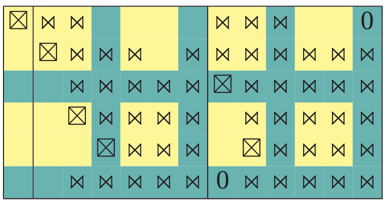
$$\downarrow G^X(2,6) \downarrow$$



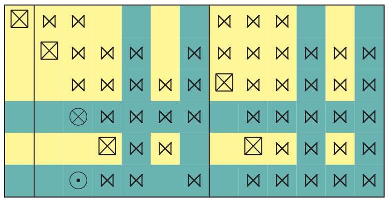
The next pivot comes from the Y part, third row.



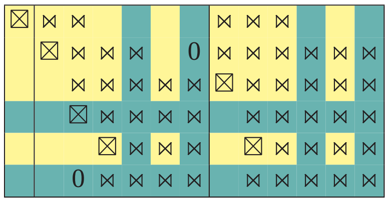
$$\downarrow G^Y(3,6) \downarrow$$



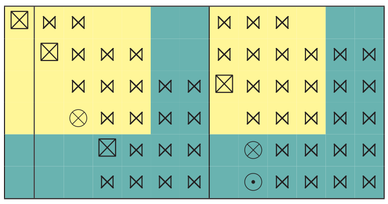
The next pivot comes from the X part, fourth row.



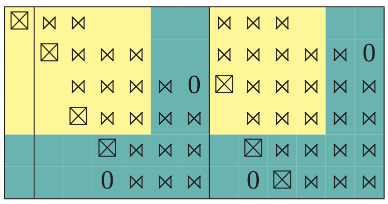
$$\downarrow G^X(4,6) \downarrow$$



The next pivot comes from the fifth row, but we have a choice between the X part and the Y part. Our ordering scheme (12) tells us to take the pivot from the Y part. However, we could also choose the pivot from the X-part, e.g. if this would enhance the numerical stability. Note that the element below the pivot position in the 5th row in X part also becomes zero. At this step we mark by $\boxtimes$ the pivot for the 6th polynomial in the Y-part, its position is in the 6th row.



$$\downarrow G^Y(5,6) \downarrow$$



The elements of the last matrix are exactly the values appearing in the recurrence system of equations (10).

5 Numerical experiments

We have implemented Algorithm 1 in Matlab and applied it to several problems.

In the implementation we use the ordering scheme (12). As the points for the discrete inner product we use Padua points. The software for working with Padua points is described in [6].

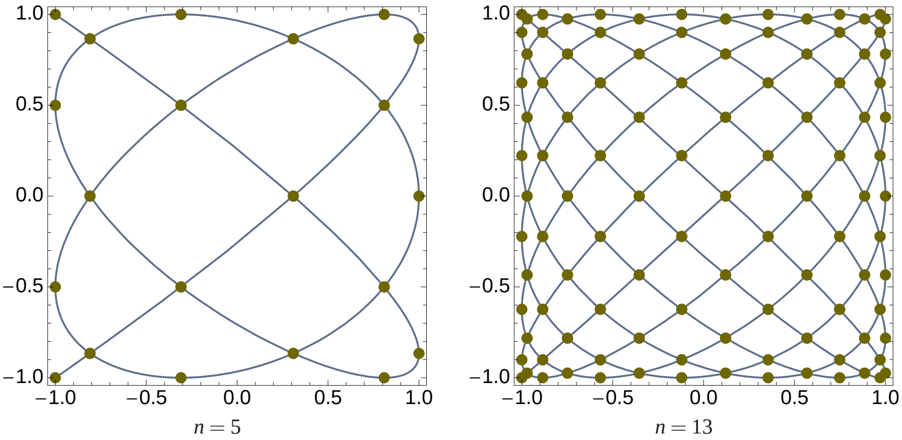


Fig. 1 Padua points for different n

The $m+1 = (\delta+1)(\delta+2)/2$ Padua points corresponding to degree $\delta > 0$ are the set of points

$$\mathrm{Pad}_\delta = \{\boldsymbol{\zeta} = (\zeta_1, \zeta_2)\} = \left\{\gamma\left(\frac{k\pi}{\delta(\delta+1)}\right), \quad k = 0,\dots,n(n+1)\right\}$$

where $\gamma(t)$ is their "generating curve"

$$\gamma(t) = (-\cos((\delta+1)t), -\cos(\delta t)) \quad t \in [0,\pi].$$

Notice that two of these points are consecutive vertices of the square, $2\delta - 1$ other points are on the edges of the square and the remaining (interior) points are corresponding to self-intersections of the generating curve, see Figure 1.

Example 1: orthogonality test

Let us recall the inverse eigenvalue problem: find a unitary matrix Q (transformation matrix) and generalized upper Hessenberg matrices H_x and H_y such that $Q^H\mathbf{w} = \|\mathbf{w}\|\mathbf{e}_1$, $Q^HXQ = H_x$ and $Q^HYQ = H_y$. Denote by $\phi_i(\zeta)$ the computed orthonormal polynomials and let $\Phi = [\phi_0(\boldsymbol{\zeta}_i)\,\phi_1(\boldsymbol{\zeta}_i)\,\dots\,\phi_m(\boldsymbol{\zeta}_i)]_{i=0}^{m+1}$, $W = \mathrm{diag}(w_i)$. Then, as it is proven before, $W\Phi = Q$. Computing the values of the OPs at the nodes $\boldsymbol{\zeta}_i$ by means of the recurrence relations based on H_x, H_y gives us a possibility to check numerically the orthogonality of the matrix $W\Phi$.

As the nodes $\boldsymbol{\zeta}_i$ we take $N = 5151$ Padua points of degree $n = 100$, and the identity matrix as the weight matrix. The values of the OPs are stored in the matrix $V = W\Phi$ and we denote by $R = |V^HV - I|$ (modulus is taken elementwise). In Figure 2 we plot $\max R(1:k, 1:k)$ for $k = 10:100:N$.

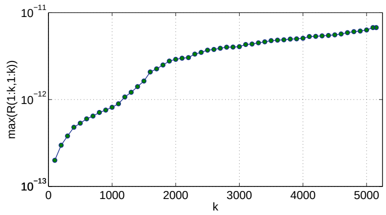


Fig. 2 Max orthogonality error for the first k OPs, $N = 5151$ points

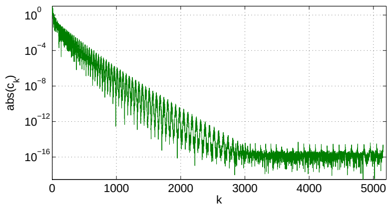


Fig. 3 LS solution coefficients for Franke function, $N = 5151$ points

Example 2: least squares solution

Recall that the solution $p(z)$ to the least squares (LS) problem (2) is $p(\zeta) = \sum_{j=0}^{n} c_j \phi_j(\zeta)$. Then $\mathbf{c} = (c_j)$ is given by $\mathbf{c} = \Phi^H W^H W \mathbf{f} = [\langle \phi_i, \mathbf{f} \rangle]$, where the matrices Φ and W are defined in the previous example, so we perform the same row operations on $W\mathbf{f}$ as on $\mathbf{w}$ in Algorithm 1.

As the test function we consider the Franke function

$$F(x,y) = \frac{3}{4} e^{-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}} + \frac{3}{4} e^{-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}} + \frac{1}{2} e^{-\frac{(9x-7)^2}{4} - \frac{(9y-3)^2}{4}} - \frac{1}{5} e^{-(9x-4)^2 - (9y-7)^2}$$

on $[0,1] \times [0,1]$ and transform the $N = 5151$ Padua points from $[-1,+1]^2$ to $[0,1]^2$. These transformed points are denoted by $\boldsymbol{\zeta}_i$. Then we compute the right-hand side $\mathbf{f} = F(\boldsymbol{\zeta}_i)$ and the least squares solution coefficients $\Phi^H W^H W \mathbf{f} = \mathbf{c}$. In Figure 3 the absolute values of the solution coefficients $\mathbf{c}_k$ are plotted for all k. It is easy to see that from $k \approx 3000$ they are of machine-precision size.

Example 3: Polynomial that goes through some points

Consider the square $[0,1] \times [0,1]$. As points, we take 20 equidistant points on the circle with center $(0.25\,;\,0.25)$ and radius 0.15. The next 20 points are taken similarly on a circle with center $(0.75\,;\,0.75)$ and radius 0.15. The last 4 points are the 4 edges of the square. We look for the polynomial having “least degree” that has zero value in the given points. This situation corresponds to the first zero pivot appearing in the recurrence relation. It happens for the 28th orthogonal polynomial. It is the polynomial of degree 6, which corresponds with the theoretical estimate (degree 2+2+1+1, two circles and two lines). Figure 5 shows the surface plot of this polynomial.

6 Conclusions

We presented an algorithm computing the recurrence relation coefficients for bivariate polynomials, orthonormal with respect to a discrete inner product. To do so, we transformed the original problem to an inverse eigenvalue problem and solved it by applying a sequence of specially built Givens rotations. The algorithm is a basis tool

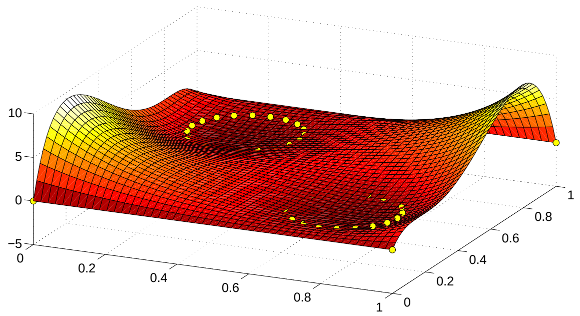


Fig. 4 Surface plot of an interpolating polynomial

in solving different problems of numerical mathematics, such as the polynomial interpolation problem or the discrete least squares problem. Numerical examples show that the algorithm could indeed be efficiently applied to such problems and also illustrate its good numerical stability.

References

1. G. S. Ammar, W. B. Gragg, and L. Reichel, *Constructing a unitary Hessenberg matrix from spectral data*, in Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms, G. H. Golub and P. Van Dooren, eds., vol. 70 of Computer and Systems Sciences, Springer-Verlag, Berlin, Germany, 1991, pp. 385–395.
2. G. S. Ammar and C. He, *On an inverse eigenvalue problem for unitary Hessenberg matrices*, Linear Algebra Appl., 218 (1995), pp. 263–271.
3. L. Bos, M. Caliari, S. De Marchi, M. Vianello, and Y. Xu, *Bivariate Lagrange interpolation at the Padua points: the generating curve approach*, J. Approx. Theory, 143 (2006), pp. 15–25.
4. L. Bos, S. De Marchi, M. Vianello, and Y. Xu, *Bivariate Lagrange interpolation at the Padua points: the ideal theory approach*, Numer. Math., 108 (2007), pp. 43–57.
5. A. Bultheel and M. Van Barel, *Vector orthogonal polynomials and least squares approximation*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 863–885.
6. M. Caliari, S. De Marchi, A. Sommariva, and M. Vianello, *Padua2DM: fast interpolation and cubature at the Padua points in Matlab/Octave*, Numer. Algorithms, (2009). Submitted.
7. M. Caliari, S. De Marchi, and M. Vianello, *Bivariate polynomial interpolation on the square at new nodal sets*, Appl. Math. Comput., 165 (2005), pp. 261–274.
8. M. T. Chu and G. H. Golub, *Inverse Eigenvalue Problems: Theory, Algorithms and Applications*, Numerical Mathematics & Scientific Computations, Oxford University Press, New York, USA, 2005.
9. S. Elhay, G. H. Golub, and J. Kautsky, *Updating and downdating of orthogonal polynomials with data fitting applications*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 327–353.
10. H. Fassbender, *On numerical methods for discrete least-squares approximation by trigonometric polynomials*, Math. Comp., 66 (1997), pp. 719–741.
11. W. Gautschi, *Orthogonal Polynomials: Computation and Approximation*, Oxford University Press, New York, USA, 2004.
12. G. H. Golub and G. Meurant, *Matrices, Moments and Quadrature with Applications*, Princeton Series in Applied Mathematics, Princeton University Press, 2009.
13. L. Reichel, *Fast QR-decomposition of Vandermonde-like matrices and polynomial least squares approximation*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 552–564.
14. L. Reichel, G. S. Ammar, and W. B. Gragg, *Discrete least squares approximation by trigonometric polynomials*, Math. Comp., 57 (1991), pp. 273–289.
15. M. Van Barel and A. Bultheel, *Orthonormal polynomial vectors and least squares approximation for a discrete inner product*, Electron. Trans. Numer. Anal., 3 (1995), pp. 1–23.