On Polynomial and Polynomial Matrix Interpolation

Petr Hušek and Renata Pytelková

Department of Control Engineering, Faculty of Electrical Engineering Czech Technical University in Prague Technická 2, 166 27, Praha 6, Czech Republic {husek,dvorako}@fel.cvut.cz

Abstract. The classical algorithms for computations with polynomials and polynomial matrices use elementary operations with their coefficients. The relative accuracy of such algorithms is relatively small and for polynomials of higher order and polynomial matrices of higher dimension the executing time grows very quickly. Another possibility is to use symbolic manipulation package but even this is applicable only for moderate problems. This paper improves a new method based on polynomial interpolation. Its principle is as follows [1]: firstly a sufficient number of interpolation points is chosen, then the interpolated object is evaluated in these points and finally it is recovered from both series of values. The choice of interpolation points is crucial to have a well-conditioned task. Typically, a random choice of real points leads to a badly conditioning for higher order of interpolated polynomial. However, a set of complex points regularly distributed on the unit circle in the complex plane gives a perfectly conditioned task. Moreover very efficient algorithm of fast Fourier transform can be used to recover the resulted polynomial or polynomial matrix. The efficiency is demonstrated on determination of inverse to polynomial matrix.

1 Introduction

Solution of many technical problems leads to computations with polynomials or polynomial matrices. The classical algorithms, which are based on manipulations with coefficients, are distinguished for small relative accuracy and low efficiency. The algorithms based on interpolation-evaluation techniques represent a computationally efficient way to deal with such computations. The interpolation theory for polynomial case is very old. Nevertheless even the problem is very well studied all the algorithms meet the problem of badly conditioned task and could be used only for polynomials of lower degree (up to 30). The generalization to the matrix case appeared only recently. The most general way, which includes all the interpolation schemes, was introduced by Antsaklis [1], generalizing the approach [5]. However, the problem of badly conditioned task remained unsolved even if some ways for its removing were suggested. The different polynomial bases were used in Lagrange and Hermite interpolation ([6]).

The computational aspects of the enhancement to multivariate case are summarized in [3] and [4].

In this paper the conditioned number of corresponding task is improved by choosing a special set of interpolation points in both, polynomial and polynomial matrix case [7,8]. Moreover, the fast Fourier transform (FFT) algorithm is used for evaluation of the interpolated object in the prescribed set of points and its recovery from these values. The proposed algorithm is very efficient and numerically reliable.

2 Polynomial Case

Firstly let us remind, that the term *interpolation* can be used to denote many different mathematical tools. The technique, which will be here understood as interpolation, is described below. The basic idea of all interpolation methods can be summarized in the three following steps:

- 1. A sufficient number of interpolation points is chosen.
- 2. The interpolated object is evaluated in this set of points.
- 3. The interpolated object is recovered from both series of values.

The simplest example of interpolation technique is univariate polynomial interpolation. The corresponding procedure can be stated by the following theorem.

THEOREM 1 (Univariate polynomial interpolation [1])

Given K distinct generally complex scalars s_j , j = 1, ..., K, and K corresponding complex values b_j , there exists a unique polynomial r(s) of degree n = K - 1 for which

$$r(s_{i}) = b_{i}, \quad j = 1, \dots, K$$

That is, an *n*th degree polynomial r(s) can be uniquely represented by the K = n+1 interpolation points (or pairs) (s_j, b_j) , j = 1, ..., K. To see this, the *n*th degree polynomial $r(s) = r_0 + r_1 s + \dots + r_n s^n$ can be written as $r(s) = \mathbf{r} [1, s, \dots, s^n]^T$ where $\mathbf{r} = [r_0, r_1, \dots, r_n]$ is $(1 \times (n+1))$ row vector of the coefficients. The K = n+1 equations can then be written as

$$\mathbf{r}\mathbf{V} = \mathbf{r} \begin{bmatrix} 1 & \dots & 1 \\ s_1 & & s_K \\ \vdots & & \vdots \\ s_1^{K-1} & \dots & s_K^{K-1} \end{bmatrix} = \begin{bmatrix} b_1, \dots, b_K \end{bmatrix} = \mathbf{b} .$$
⁽²⁾

The matrix **V** is called Vandermonde matrix and it is non-singular if and only if the *K* scalars s_j , j = 1, ..., K are distinct. In that case the equation (2) has a unique solution **r**; that is, there exists a unique polynomial r(s) of degree *n* which satisfies (1). This proves the theorem 1.

From numerical point of view the solution of equation (2) depends on the condition number of matrix V. This problem is studied in the next section.

3 Condition Number of Vandermonde Matrix

It is well-known fact that the Vandermonde matrix (2) for vector of real numbers is very badly conditioned. Typically, for interpolation points regularly distributed in the interval [-1,1] the critical MATLAB condition number 10^{16} is achieved for K = 30. An immense improvement of the condition number of the Vandermonde matrix can be surprisingly achieved using complex interpolation points even if it seems to be contraproductive.

Let us choose the vector of interpolation points as

$$[s_1, \dots, s_K] = [1, \omega, \omega^2, \dots, \omega^{K-1}] / \sqrt{K}$$
⁽⁵⁾

(2)

(4)

where ω denotes a primitive *K*-th root of 1, that is

$$\omega^{K} = 1, \quad \omega^{h} \neq 1, \quad h = 1, \dots, K - 1.$$

Such a set of interpolation points is called Fourier points. The corresponding Vandermonde matrix Ω (referred to as Fourier matrix) has the following form:

$$\boldsymbol{\Omega} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \boldsymbol{\omega} & \boldsymbol{\omega}^2 & \dots & \boldsymbol{\omega}^{K-1} \\ 1 & \boldsymbol{\omega}^2 & \boldsymbol{\omega}^4 & \dots & \boldsymbol{\omega}^{2(K-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \boldsymbol{\omega}^{K-1} & \boldsymbol{\omega}^{2(K-1)} & \dots & \boldsymbol{\omega}^{(K-1)(K-1)} \end{bmatrix} / \sqrt{K} .$$
(5)

Observe that since $\sum_{i=0}^{K-1} (\omega \widetilde{\omega})^{ih} = K - 1$ for $1 \le h \le K$ where $\widetilde{\omega} = 1/\omega$ is a primitive root of 1, the Euclidean norm of each column appearing in the matrix Ω is equal to \sqrt{K} . Moreover since $\sum_{i=0}^{K} \omega^{hi} = 0$ for $1 \le h \le K$, the matrix Ω is unitary and

$$\operatorname{cond}(\Omega) = 1.$$
 (0)

It means that the Fourier matrix Ω is perfectly conditioned and its inverse can be determined as its complex conjugate (7)

$$\Omega^{-1} = \Omega^*.$$

The vector of coefficients of interpolated polynomial can be then obtained as the solution of (2) as

$$\mathbf{r} = \mathbf{b} \mathbf{\Omega}^* / \sqrt{K} = \mathbf{b} \left[\widetilde{\omega}_{ij} \right] / K .$$

Another important fact is that in order to evaluate a univariate polynomial in the Fourier points, that corresponds to step 1 of the general interpolation-evaluation algorithm mentioned above, the fast Fourier transform (FFT) algorithm could be applied. The algorithm is well studied, very efficient and numerically reliable. Moreover, as the Fourier matrix is unitary, the inverse FFT algorithm can be used to

recover the polynomial coefficients from its values (step 3 of that algorithm or equivalently the product in (8)).

The use of the described algorithm will be shown on an example in section 6 on determination of determinant of polynomial matrix.

In the following section a generalization of the algorithm described above to polynomial matrix case is introduced.

4 Polynomial Matrix Case

The result obtained above can be generalized to polynomial matrix case in many different ways. Nevertheless all of them can be seen as special cases of the basic polynomial matrix interpolation procedure introduced by the following theorem.

Let $\mathbf{S}(s) := \text{block diag}\{[1, s, ..., s^{d_i}]^T\}$ where $d_i, i = 1, ..., m$ are non-negative integers; let $\mathbf{a}_j \neq 0$ and \mathbf{b}_j denote $(m \times 1)$ and $(p \times 1)$ complex vectors respectively and s_i complex scalars.

THEOREM 2 (Polynomial matrix interpolation [3])

Given interpolation (points) triplets $(s_j, \mathbf{a}_j, \mathbf{b}_j)$, j = 1, ..., K and non-negative integers d_i , i = 1, ..., m with $K = \sum d_i + m$ such that the $(\sum d_i + m) \times K$ matrix

$$\mathbf{S}_{K} := [\mathbf{S}(s_{1})\mathbf{a}_{1}, \dots, \mathbf{S}(s_{K})\mathbf{a}_{K}]$$
⁽⁹⁾

has full rank, there exists a unique $(p \times m)$ polynomial matrix $\mathbf{Q}(s)$, with *i*th column degree equal to d_i , i = 1, ..., m for which

$$\mathbf{Q}(s_j)\mathbf{a}_j = \mathbf{b}_j, \, j = 1, \dots, K \; .$$

 $\mathbf{Q}(s)$ can be written as

$$\mathbf{Q}(s) = \mathbf{Q}\mathbf{S}(s) \tag{11}$$

(10)

(11)

where **Q** $(p \times (\sum d_i + m))$ contains the coefficients of the polynomial entries. **Q** must satisfy

$$\mathbf{QS}_{\mathcal{K}} = \mathbf{B}_{\mathcal{K}}$$
(12)

where $\mathbf{B}_{K} = [\mathbf{b}_{1}, \dots, \mathbf{b}_{K}]$. Since \mathbf{S}_{K} is non-singular, \mathbf{Q} and therefore $\mathbf{Q}(s)$ are uniquely determined.

For p = m = 1 **S**_{*K*} is turned to *Vandermonde matrix*. In the polynomial matrix case, we shall call it *block Vandermonde matrix* because of its structure, that will be shown later.

Both theorems expand step 3 in the general description of interpolation methods introduced before. They show, that the resulted polynomial or polynomial matrix can be found as a solution of the matrix equation (6) or (8) respectively. It means, that the step 3 is the same for every problem solvable by interpolation techniques. The whole difference consists in step 2, which of course can be performed by algorithms

working with constant matrices. These algorithms are usually very efficient and therefore guarantee good numerical properties of results computed in a short time.

5 Block Vandermonde matrix

Let us focus on the block Vandermonde matrix S_K appearing in (12) defined by (9). According to theorem 2 the choice of interpolation triplets $(s_j, \mathbf{a}_j, \mathbf{b}_j)$ is arbitrary. The only condition is the full rank of matrix S_K . The postmultiplication of interpolated matrix by the vectors \mathbf{a}_j makes it possible to vary interpolation triplets not only by points s_j but also by the vectors \mathbf{a}_j . This is compensated by a higher number of points than they are necessary for element by element interpolation procedure. In section 3 it was shown that Vandermonde matrix has some appropriate properties if evaluated in the Fourier points. Let us try to choose such triplets $(s_j, \mathbf{a}_j, \mathbf{b}_j)$ to maintain this structure in the block Vandermonde matrix.

Let $K = \sum d_i + m$, j = 1,...,K, i = 1,...,m, $d_0 = 0$, q = 1,...,r, $r = 1 + \max d_i$:

let
$$s_j = s_q$$
, $\mathbf{a}_j = \mathbf{e}_i$ for $j \in \left[\sum_{k=0}^{i-1} d_k + i, \sum_{k=1}^{i} d_k + i\right];$ (13)
 $q = j - \sum_{k=1}^{i-1} d_k - i + 1, \ \mathbf{e}_i = [0, \dots, 0, 1, 0, \dots, 0]$

where 1 is on the *i*th position.

This set of interpolation triplets requires only *r* different interpolation points s_q , which is the lowest possible number and it corresponds to the column-wise interpolation procedure. The block Vandermonde matrix S_K has the following form:

$$\mathbf{S}_{K} = \begin{bmatrix} \mathbf{V}_{1} & 0 & \cdots & 0 \\ 0 & \mathbf{V}_{2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{V}_{m} \end{bmatrix}$$
(14)

where each V_i , i = 1, ..., m is the Vandermonde matrix

$$\mathbf{V}_{i} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ s_{1} & s_{2} & \cdots & s_{d_{i}} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1}^{d_{i}} & s_{2}^{d_{i}} & \cdots & s_{d_{i}}^{d_{i}} \end{bmatrix}.$$
 (15)

From (14) it immediately follows that the inverse to S_K appearing in solution of (12) can be determined as

$$\mathbf{S}_{K}^{-1} = \begin{bmatrix} \mathbf{V}_{1}^{-1} & 0 & \cdots & 0 \\ 0 & \mathbf{V}_{2}^{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{V}_{m}^{-1} \end{bmatrix}.$$
 (16)

If the interpolation points s_i are chosen as the Fourier ones then $\mathbf{S}_K = \text{diag}\{\mathbf{\Omega}_i; i = 1, ..., m\}$ and $\mathbf{S}_K^{-1} = \text{diag}\{\mathbf{\Omega}_i^*; i = 1, ..., m\}$ where

$$\boldsymbol{\Omega}_{i} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \boldsymbol{\omega} & \boldsymbol{\omega}^{2} & \dots & \boldsymbol{\omega}^{d_{i}} \\ 1 & \boldsymbol{\omega}^{2} & \boldsymbol{\omega}^{4} & \dots & \boldsymbol{\omega}^{2d_{i}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \boldsymbol{\omega}^{d_{i}} & \boldsymbol{\omega}^{2d_{i}} & \dots & \boldsymbol{\omega}^{d_{i}d_{i}} \end{bmatrix} / \sqrt{d_{i} + 1} .$$

$$(17)$$

Again the fast Fourier transform algorithm can be used for evaluation of $\mathbf{Q}(s)$ in these points and its recovery from them.

6 Example

Let us use the described procedure to determine inverse to polynomial matrix A(s),

$$\mathbf{A}(s) = \begin{bmatrix} s^2 + 1 & s & s + 2 \\ s^2 & 1 & 2s^2 + 3 \\ 0 & s^2 + 2s + 2 & 2s + 1 \end{bmatrix}.$$

The inverse to polynomial matrix A(s) can be determined as

$$\mathbf{A}(s)^{-1} = [\det(\mathbf{A}(s))]^{-1} \operatorname{adj}(\mathbf{A}(s)) = r(s)^{-1} \mathbf{Q}(s)$$

where adj stands for adjoint to a matrix.

Firstly let us compute the determinant r(s) using procedure described in section 2.

The first step consists in estimation of degree *n* of *r*(*s*). It is important to note that if the estimated degree is higher than the real one, the coefficients corresponding to high powers will be determined as zero because due to theorem 1 there is only one polynomial of *n*th degree determined by n + 1 points (in fact it should be "of *n*th or less degree"). Obviously, the estimated degree cannot be less than the real one. The simplest choice is $n = \sum_{i=1}^{m} c_i$, where c_i are the column degrees of **A**(*s*) (the highest degrees appearing in the column).

Here n = 6 and the Fourier matrix Ω is of the seventh (K = n + 1 = 7) order. Now we have to evaluate matrix $\mathbf{A}(s)$ in the interpolation points $[1, \omega, ..., \omega^6]$ where ω is seventh root of 1. This step is performed by FFT algorithm. Next the vector of values

of the determinants $\mathbf{b} = [\det(\mathbf{A}(1)), \det(\mathbf{A}(\omega)), \dots, \det(\mathbf{A}(\omega^6))]$ is computed using standard procedure for computing determinant of a constant matrix. The recovery of all the coefficients of r(s) (solution of (8)) is performed via inverse FFT applied on the vector **b**.

The algorithm gives $r(s) = -2s^6 - 3s^5 - 7s^4 - 3s^3 - 8s^2 - 4s - 5$. Now let us determine the polynomial matrix $\mathbf{Q}(s) = \operatorname{adj}(\mathbf{A}(s))$.

As in the previous procedure firstly the column degrees d_i of $\mathbf{Q}(s)$ have to be estimated. The simplest choice is $d_i = (m-1)c, i = 1, ..., m$ where $c = \max c_i$. In our case $d_i = 4, i = 1, ..., 3$. Let us choose the interpolation points according to (13) as the Fourier points. We need $r = \max d_i + 1 = 5$ scalar complex points $s_j, j = 1, ..., 5$, $[s_1, ..., s_5] = [1, \omega, ..., \omega^4]$ where ω is the fifth root of 1. The total number of interpolation triplets is $K = \sum_{i=1}^m d_i + m = 15$. The procedure continues as follows. The polynomial matrix $\mathbf{A}(s)$ is evaluated in the points $s_j, j = 1, ..., 5$. Next the values of $\mathbf{Q}(s)$ are computed in those points: $\mathbf{Q}(s_j) = \operatorname{adj}(\mathbf{A}(s_j)), j = 1, ..., 5$. Finally, the solution of (12) can be determined by applying inverse FFT algorithm on each entry of $\mathbf{Q}(s_j)$.

The algorithm gives

$$\mathbf{Q}(s) = \begin{bmatrix} -2s^4 - 4s^3 - 7s^2 - 4s - 5 & s^3 + 2s^2 + 5s + 4 & 2s^3 + 2s - 2 \\ -2s^3 - s^2 & 2s^3 + s^2 + 2s + 1 & -2s^4 + s^3 - 3s^2 - 3 \\ s^4 + 2s^3 + 2s^2 & -s^4 - 2s^3 - 3s^2 - 2s - 2 & -s^3 + s^2 + 1 \end{bmatrix}$$

7 Conclusion

In the paper the general concept of algorithms on polynomials and polynomial matrices based on interpolation-evaluation methods was presented. The algorithms are very efficient and numerically reliable because of using the FFT algorithm, very well studied algorithms for constant matrices and perfectly conditioned task. The method was used for computing inverse to polynomial matrix. The experiments reveal that the algorithm is able to deal with polynomial matrices of high dimension with elements of high degrees. For example, determinant of a polynomial matrix (30×30) with elements of degrees 30 is computed in 2 seconds¹. Adjoint to the same matrix is determined in 10 seconds. The only drawback of interpolation-evaluation methods consists in big storage capacity of a computer needed to store all the values of interpolated object in interpolation points for some algorithms. For example, to determine the determinant above one need to store 810900 complex values (even if the determinant is of degree 401 only). However, to determine the adjoint above the

¹ All the computations was done on Pentium II, 64MB, 120MHz.

same number of values has to be stored as the total number of coefficients of the adjoint.

Acknowledgements

This work has been supported by the research program No. J04/98:212300013 "Decision Making and Control for Manufacturing" of the Czech Technical University in Prague (sponsored by the Ministry of Education of the Czech Republic).

References

- 1. Antsaklis, P.J. and Gao, Z.: Polynomial and Rational Matrix Interpolation: Theory and Control Applications. *International Journal of Control*. Vol. 58, No. 2, pp.349-404 (1993)
- 2. Bini, D. and Pan, V.: Polynomial and Matrix Computations, Vol. 1, *Fundamental Algorithms*. Birkhäuser, Boston (1994)
- 3. De Boor, C. and Ron, A.: On Multivariate Polynomial Interpolation. *Constr. Approx.*, **6**, pp.287-302 (1990)
- De Boor, C. and Ron, A.: Computational Aspects of Polynomial Interpolation in Several Variables. Math. Comp., 58, pp.705-727 (1992)
- 5. Davis, P.J.: Interpolation and Approximation. Dover Books, 2.edition, (1975)
- 6. Gasca, M. and Maeztu, J.I. On Lagrange and Hermite interpolation in \Re^k . Numer. Math., 39, pp.1-14 (1982)
- Hušek, P. and Dvořáková, R.: Stability Analysis of Takagi-Sugeno Fuzzy Systems with Linear Input-Output Submodels: Polynomial Approach, in: Proc. of 11th IFAC Workshop Control Applications of Optimization (CAO 2000), St. Petersburg, Russia (2000)
- 8. Hušek, P. and Štecha, J.: Rational Interpolation. In: *Proceedings of the 1st Europoly Workshop Polynomial Systems Theory & Applications*. Glasgow. Europoly-the European Network of Excellence no. CP97-7010, pp.121-126 (1999)