Applied Mathematics and Computation 218 (2011) 2044–2054

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Numerical recipes for the high efficient inverse of the confluent Vandermonde matrices

Jerzy Stefan Respondek *

Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, Institute of Computer Science, ul. Akademicka 16, 44-100 Gliwice, Poland
Higher School of Finance and Law in Bielsko-Biała, ul. Tańskiego 5, 43-382 Bielsko-Biała, Poland

ARTICLE INFO

Keywords:
Numerical recipes
Numerical algebra
Linear algebra
Matrix inversion
Confluent Vandermonde matrices
C++

ABSTRACT

In this paper, we derive a numerical recipe for the calculation of the inversion of the confluent Vandermonde matrix. The main result of this article does not require any symbolic calculations and therefore can be performed by a numerical algorithm implemented either in any high level (like Matlab or Mathematica) or low level programming language (C/C++/Java/Pascal/Fortran, etc.). The computational complexity of the presented algorithm is of an $O(N^2)$ class being, by the linear term, better than the ordinary Gauss elimination method. Moreover, a ready to use C++ full implementation of the algorithm is attached.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

In this paper, we derive a numerical algorithm for the calculation of the inversion of the confluent Vandermonde matrix. The main result of this article is presented by Theorems 5.1 and 6.1. It does not require any symbolic calculations and therefore can be performed by a numerical algorithm implemented in any high (like Matlab or Mathematica) or low level programming language (C/C++/Java/Pascal/Fortran, etc.).

The paper is organized as follows: Section 2 justifies the importance of the confluent Vandermonde matrices, in Section 3 we performed a comparison with other methods, Section 4 provides the necessary theoretical background, Section 5 – the algorithm for the auxiliary coefficients calculation, Section 6 – the final algorithm for calculating the desired inversion, in Section 7 we determined the computational complexity of the presented algorithm, in Section 8 we presented the sketch of the proposed algorithm, in Section 9 we give an example from real application, in Section 10 we tested the behavior for the poorly conditioned matrices and in Section 11 we summarized the article. In the appendix we presented a ready to use C++ implementation of the algorithm.

2. Importance of the confluent Vandermonde matrices

The confluent Vandermonde matrices we formally defined in Section 4. They arise in a broad range of both theoretical and practical problems. Below we surveyed the problems which make it necessary to inverse the confluent Vandermonde matrices.

- Linear, ordinary differential equations: the Jordan canonical form matrix of the ODE in the Frobenius form is a confluent Vandermonde matrix [3, pp. 86–95].

* Corresponding author at: Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, Institute of Computer Science, ul. Akademicka 16, 44-100 Gliwice, Poland.
E-mail address: Jerzy.Respondek@polsl.pl

0096-3003/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2011.07.017

- Control problems: investigating of the so-called controllability [7] of the higher order systems leads to the problem of inverting the classic Vandermonde matrix [11] (in case of distinct zeros of the system characteristic polynomial) and the confluent Vandermonde matrix [12] (for systems with multiply characteristic polynomial zeros). As the examples of the higher order models of the physical objects may be mentioned the Timoshenko's elastic beam equation [15] (4th order) and the Korteweg-de Vries equation of waves on shallow water surfaces [1] (3rd, 5th and 7th order).
- Interpolation: apart from the ordinary polynomial interpolation with single nodes we consider the Hermite interpolation, allowing the multiple interpolation nodes. That problem leads to the system of linear equations, with the confluent Vandermonde matrix [6, pp. 363–373].
- Information coding: the confluent Vandermonde matrix is used in coding and decoding the information in the Hermitian's code [8].
- Optimization of the non-homogeneous differential equations [2].

3. Comparison with other methods

In the literature the problem of inverting the confluent Vandermonde matrices has been tackled several times:

- The article [5] proposes a method requiring hand-held calculation of the fractional expansion of the inverse of the characteristic polynomial. Moreover, the numerical part of the proposed method is of the weak $O(N^4)$ class.
- The method proposed in [4] optimizes the numerical part of the article [5], but still requires performing the same symbolic fractional expansion.
- The work [13] requires computing the arbitrary order derivatives of the inverse characteristic polynomial.

Summarizing, previous results on this topic cannot be performed numerically as they require either manual calculation of a series of the derivatives (3) or the knowledge of the fractional expansion of the inverse of the characteristic polynomial.

4. Theoretical background

First let us explain what the confluent Vandermonde matrix V is. Let $\lambda_1, \lambda_2, \ldots, \lambda_r$ be given pair wise distinct zeros of the polynomial $p(s) = (s-\lambda_1)^{n_1} \cdots (s-\lambda_r)^{n_r}$ with $n_1 + \cdots + n_r = n$. The confluent Vandermonde matrix V related to the zeros of $p(s)$ is defined to be the $n \times n$ matrix $V = [V_1 V_2 \cdots V_r]$, where the block matrix $V_k = V(\lambda_k, n_k)$ is of order $n \times n_k$ having elements [4,13]

$$[V(\lambda_k, n_k)]_{ij} = \begin{cases} \binom{i-1}{j-1} \lambda_k^{i-j}, & \text{for } i \geqslant j, \\ 0, & \text{otherwise} \end{cases}$$

for $k = 1,2,\ldots,r$; $i = 1,2,\ldots,n$ and $j = 1,2,\ldots,n_k$.

The articles [4,13] present the following theorem for inverting the confluent Vandermonde matrices:

Theorem 4.1. *The inverse of the confluent Vandermonde matrix V has the form $V^{-1} = \begin{bmatrix} W_1^T & W_2^T & \cdots & W_r^T \end{bmatrix}^T$. The column vectors h_{ki} of the block matrix $W_k = [h_{kn}, \ldots, h_{k1}]$ in the inverse confluent Vandermonde matrix V^{-1} may be recursively computed by the following scheme:*

$$\begin{cases} h_{k1} = [K_{k,1} \cdots K_{k,n_k}]^T, \\ h_{k2} = J_k(\lambda_k, n_k) h_{k1} + a_1 h_{k1}, \\ \vdots \\ h_{kn} = J_k(\lambda_k, n_k) h_{k(n-1)} + a_{n-1} h_{k1}, \end{cases} \qquad k = 1, 2, \ldots, r, \tag{1}$$

where λ_k is the eigenvalue, a_k are the coefficients of the characteristic polynomial, $J_k(\lambda_k, n_k)$ is the elementary Jordan block (2):

$$J_k(\lambda_k, n_k) = \begin{bmatrix} \lambda_k & 1 & & \cdots & 0 \\ & \lambda_k & 1 & & \vdots \\ & & \ddots & \ddots & \\ & & & \lambda_k & 1 \\ 0 & & & & \lambda_k \end{bmatrix}_{n_k \times n_k}, \quad k = 1, 2, \ldots, r \tag{2}$$

and K_{kj} are the auxiliary coefficients expressed by the formula (3):

$$K_{kj}(s)|_{s=\lambda_k} = \frac{1}{(n_k - j)!} \frac{d^{(n_k-j)}}{ds^{(n_k-j)}} \left[\frac{1}{p(s)} (s-\lambda_k)^{n_k} \right]_{s=\lambda_k}, \begin{matrix} j = n_k, \ldots, 1, \\ k = 1, 2, \ldots, r. \end{matrix} \tag{3}$$

Download English Version:

https://daneshyari.com/en/article/4631583

Download Persian Version:

https://daneshyari.com/article/4631583

Daneshyari.com