Monografías Matemáticas *García de Galdeano* **40**, 53–60 (2015)

FACTORIZATION OF VANDERMONDE MATRICES AND THE NEWTON FORMULA

Jesús Carnicer, Yasmina Khiar and Juan Manuel Peña

Abstract. We revisit the connection between the Lagrange polynomial interpolation problem and the linear systems of equations with Vandermonde coefficient matrices. We present several related algorithms that can be computed to high relative accuracy (HRA). In fact, we propose an algorithm to compute the triangular matrices of the Crout factorization and their inverses with high relative acurracy.

Keywords: Vandermonde matrix, *LU* factorization, high relative accuracy.

AMS classification: 15A23, 65F35, 41A05.

§1. Introduction

In the last years, the research on algorithms with HRA is being a very active field in Numerical Linear Algebra. These algorithms have been obtained for some structured classes of matrices. One of these classes of matrices is given by the totally positive matrices, which are matrices with all their minors nonnegative. Nonsingular totally positive matrices can be factorized as a product of nonnegative bidiagonal matrices. This factorization is unique under certain restrictions (see [4]) and provides a natural parameterization of nonsingular totally positive matrices. In [7] it was shown that, if we know the bidiagonal factorization of a nonsingular totally positive matrix with HRA, then many algebraic problems associated with these matrices can be calculated with HRA, including the computation of its eigenvalues and singular values, its inverse or its triangular factorization.

The Vandermonde matrix is the coefficient matrix of a linear system for solving a Lagrange interpolation problem. The interpolation points are usually called nodes. A Vandermonde matrix with increasing nonnegative nodes is nonsingular totally positive and in [3] it was proved that the mentioned bidiagonal decomposition can be obtained with HRA. In contrast to the previous result, in this paper we present algorithms with HRA even for Vandermonde matrices that are not totally positive. Further results on the stability of linear systems with Vandermonde matrices can be found in [5], [6] and on the stability of totally positive linear systems can be found in [1] and [8].

There are several triangular factorizations for a nonsingular matrix. The Crout factorization decomposes the matrix as a product of a lower triangular matrix and an upper triangular matrix with unit diagonal. The Crout factorization of a Vandermonde matrix is closely related with the Newton interpolation formula. The stability of the Newton formula might depend on the order of the nodes (see [9]). Therefore it is important to consider different orderings of the nodes in a Vandermonde matrix as we perform in this paper.

The layout of this paper is as follows. Section 2 relates the Lagrange interpolation formula with the inverse of a Vandermonde matrix. Section 3 shows how to obtain the Crout factorization, $V = LU$, of a Vandermonde matrix using the Newton interpolation formula. In

Section 4 it is suggested an algorithm for computing the inverses L^{-1} and U^{-1}. Finally, in Section 5 we apply the previous results to compute V^{-1}, L, L^{-1}, U and U^{-1} with HRA.

§2. Lagrange formula and the inverse of a Vandermonde matrix

A classical problem in approximation theory is the *Lagrange interpolation problem.*

Lagrange interpolation problem. Let U be an $(n+1)$-dimensional function space. Given distinct nodes $x_0, \dots, x_n$ and given values $f_0, \dots, f_n$, find $u \in U$ such that $u(x_i) = f_i$, $i = 0, \dots, n$.

The interpolant can be obtained in different spaces and can be expressed in terms of different bases. If $(u_0, \dots, u_n)$ is a basis of U, we can write the solution u in the form $u = \sum_{i=0}^n c_i u_i$ and the interpolation problem is reduced to solve the linear system

$$M\binom{u_0, \dots, u_n}{x_0, \dots, x_n}\mathbf{c} = \mathbf{f},$$

where $\mathbf{c} = (c_0, \dots, c_n)^T, \mathbf{f} = (f_0, \dots, f_n)^T$ in $\mathbb{R}^{n+1}$ and

$$M\binom{u_0, \dots, u_n}{x_0, \dots, x_n} = (u_j(x_i))_{i,j=0,\dots,n}$$

in $\mathbb{R}^{(n+1)\times(n+1)}$, denotes the *collocation matrix* of the basis at the nodes. Here we use the convention that the first index of rows and columns is 0 instead of 1, so that the corresponding dimension is $n+1$.

If we take U as the space of the polynomials of degree less than or equal to n, P_n, then the Lagrange interpolation problem has always a unique solution. In order to solve it, we express the interpolation polynomial respect to the monomial basis, $\mathbf{m} = (m_0, \dots, m_n)$, where $m_j(x) = x^j$, $j = 0, \dots, n$. Setting $p(x) = \mathbf{m}(x)^T\mathbf{c} = \sum_{j=0}^n c_j x^j$. Then the problem is reduced to the linear system $V\mathbf{c} = \mathbf{f}$, where

$$V = V(x_0, \dots, x_n) := M\binom{m_0, m_1, \dots, m_n}{x_0, x_1, \dots, x_n} = \begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix} \tag{1}$$

is the *Vandermonde matrix* with nodes $x_0, \dots, x_n$. Note that V is a nonsingular matrix for any distinct $x_0, \dots, x_n$ because of the unisolvence of the Lagrange polynomial interpolation problem. Thus the solution of the interpolation problem is

$$p = \mathbf{m^T c}, \quad \mathbf{c} = V^{-1}\mathbf{f}. \tag{2}$$

The functions

$$l_j(x) = \prod_{k\neq j} \frac{x - x_k}{x_j - x_k}, \quad j = 0, \dots, n,$$

form the Langrange basis $\mathbf{l} = (l_0, \ldots, l_n)^T$. It is well-known that the solution of the Lagrange polynomial interpolation problem can be expressed by the *Lagrange formula*

$$p = \mathbf{l}^{\mathbf{T}}\mathbf{f}. \tag{3}$$

From (2) and (3), it follows that $\mathbf{l}^{\mathbf{T}}\mathbf{f} = \mathbf{m}^{\mathbf{T}}V^{-1}\mathbf{f}$, and, as this relation must hold for all $\mathbf{f}$, we deduce that

$$\mathbf{l}^{\mathbf{T}} = \mathbf{m}^{\mathbf{T}}V^{-1}.$$

So the matrix of change of basis between the Lagrange basis and the monomial basis is the inverse of the Vandermonde matrix. Expanding the elements of the Lagrange basis in terms of monomials, we obtain the (i, j) entry of V^{-1}

$$v_{ij}^{(-1)} = \frac{(-1)^{n-i}\sum_{k_1<\cdots<k_{n-i}\in\{0,\ldots,n\}\setminus\{j\}} x_{k_1}\cdots x_{k_{n-i}}}{\prod_{k\neq j}(x_j - x_k)}. \tag{4}$$

§3. Newton formula and the Crout factorization

If we express the polynomial interpolant in terms of the Newton basis, $\boldsymbol{\omega} = (\omega_0, \ldots, \omega_n)$ with

$$\omega_j(x) = (x - x_0)\cdots(x - x_{j-1}), \quad j = 0, \ldots, n,$$

we obtain the *Newton formula*

$$p = \boldsymbol{\omega}^T\mathbf{d}, \tag{5}$$

where $\mathbf{d} = (d_0, \ldots, d_n)$, is the vector of divided differences, $d_j := [x_0, \ldots, x_j]f$, $j = 0, \ldots, n$.

The collocation matrix of the Newton basis at the nodes $x_0, \ldots, x_n$ is

$$L := M\binom{\omega_0, \ldots, \omega_n}{x_0, \ldots, x_n} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & 0 & \cdots & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ 1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & \cdots & (x_n - x_0)\cdots(x_n - x_{n-1}) \end{pmatrix}, \tag{6}$$

whose (i, j) entry is $l_{ij} = \omega_j(x_i) = \prod_{k=0}^{j-1}(x_i - x_k)$, $j \leq i$. Note that L is a lower triangular matrix because $w_j(x_i) = 0$, if $j > i$. Evaluating (3) at each node, we deduce that the vector of divided differences, $\mathbf{d}$, is the solution of the linear system $\omega(x_i)^T\mathbf{d} = f_i$, $i = 0, \ldots, n$, that is,

$$L\mathbf{d} = \mathbf{f}. \tag{7}$$

The entries of the matrix L can be computed by recurrence, leading to the following algorithm

$$\begin{aligned} &l_{i0} = 1, \quad i = 0, \ldots, n, \\ &l_{ij} = l_{i,j-1}(x_i - x_{j-1}), \quad i = j, \ldots, n, \quad j = 1, \ldots, n. \end{aligned} \tag{8}$$

If we apply the Newton formula,

$$p(x) = (\omega_0(x), \dots, \omega_n(x)) \begin{pmatrix} [x_0]f \\ [x_0, x_1]f \\ \vdots \\ [x_0, \dots, x_n]f \end{pmatrix},$$

to $f(x) = 1, x, \dots, x^n$, we obtain

$$\mathbf{m}^T = \omega^T M\binom{m_0, m_1, \dots, m_n}{[x_0], [x_0, x_1], \dots, [x_0, \dots, x_n]}. \tag{9}$$

Taking into account that $[x_0, \dots, x_i]m_j = 0$ if $i > j$, we deduce that the matrix of change of basis between the Newton basis and the monomial basis is the upper triangular matrix

$$U := M\binom{m_0, m_1, \dots, m_n}{[x_0], [x_0, x_1], \dots, [x_0, \dots, x_n]} = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 0 & 1 & [x_0, x_1]x^2 & \cdots & [x_0, x_1]x^n \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & [x_0, \dots, x_{n-1}]x^n \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}. \tag{10}$$

Applying Leibniz's rule for divided differences to $m_j(x) = xm_{j-1}(x)$, we have that

$$[x_0, \dots, x_i]m_j = x_i[x_0, \dots, x_i]m_{j-1} + [x_0, \dots, x_{i-1}]m_{j-1},$$

that is, $u_{ij} = u_{i-1,j-1} + x_i u_{i,j-1}$. This formula suggests the following algorithm for computing the entries of U

$$\begin{aligned} &u_{00} = 1, \quad u_{0j} = x_0 u_{0,j-1}, \quad j = 1, \dots, n, \\ &u_{ii} = 1, \quad u_{ij} = u_{i-1,j-1} + x_i u_{i,j-1}, \quad j = i+1, \dots, n, \quad i = 1, \dots, n. \end{aligned} \tag{11}$$

Considering the collocation matrices at $x_0, \dots, x_n$ corresponding to the bases in formula (9), we deduce that

$$M\binom{m_0, m_1, \dots, m_n}{x_0, x_1, \dots, x_n} = M\binom{\omega_0, \omega_1, \dots, \omega_n}{x_0, x_1, \dots, x_n}U,$$

which implies that

$$V = LU, \tag{12}$$

that is, the matrices L and U form the (unique) Crout factorization of the Vandermonde matrix, where L is a lower triangular matrix whose diagonal entries are the pivots of the gaussian elimination and U is an upper triangular matrix with unit diagonal.

The LU factorization is used frequently to solve linear systems. In order to solve $V\mathbf{c} = \mathbf{f}$, with $V = LU$, we consider the following two triangular systems

$$L\mathbf{d} = \mathbf{f}, \quad U\mathbf{c} = \mathbf{d}.$$

The solution of the system $V\mathbf{c} = \mathbf{f}$ is reduced to the successive resolution of these triangular systems with matrices L and U. These systems link the solution with an intermediate vector $\mathbf{d}$, the vector of the divided differences, and, therefore, they are directly related with the Newton formula.

§4. Inverse triangular matrices

Let L and U be the triangular matrices in (6) and (10) respectively. Comparing the Lagrange formula and the Newton formula we obtain

$$\mathbf{l}(x)^T \mathbf{f} = \omega(x)^T \mathbf{d}$$

and, using (7), we have the relation $\mathbf{l}(x)^T \mathbf{f} = \omega(x)^T L^{-1} \mathbf{f}$ for all $\mathbf{f} \in \mathbb{R}^{n+1}$, and so we deduce that

$$\mathbf{l}(x)^T = \omega(x)^T L^{-1}.$$

Therefore, the inverse L^{-1} is the matrix of change of basis between the Lagrange basis and the Newton basis, and it follows that

$$L^{-1} = M\Big(\begin{matrix} l_0, l_1, \ldots, l_n \\ [x_0], [x_0, x_1], \ldots, [x_0, \ldots, x_n] \end{matrix}\Big).$$

The previous formula does not provide the entries of L^{-1} explicitly, but it describes them in terms of divided differences. A second option uses the formula

$$[x_0, \ldots, x_i]f = \sum_{j=0}^{i} \frac{f(x_j)}{\prod_{k \in \{0,\ldots,i\} \setminus \{j\}} (x_j - x_k)}.$$

In order to simplify the notation, notice that

$$\omega'_{i+1}(x_j) = \prod_{k \in \{0,\ldots,i\} \setminus \{j\}} (x_j - x_k),$$

which allows us to establish the following relation between $\mathbf{d}$ and $\mathbf{f}$

$$d_i = \sum_{j=0}^{i} \frac{f_j}{\omega'_{i+1}(x_j)}, \quad i = 0, \ldots, n.$$

By (7), $\mathbf{d} = L^{-1}\mathbf{f}$, and we deduce that the (i, j) entry of the matrix L^{-1} is

$$l_{ij}^{(-1)} = \frac{1}{\omega'_{i+1}(x_j)} = \frac{1}{\prod_{k \in \{0,\ldots,i\} \setminus j} (x_j - x_k)}$$

when $j \leq i$ and 0 otherwise. Then we can compute L^{-1} as follows

$$l_{ij}^{(-1)} = -\frac{l_{i-1,j}^{(-1)}}{x_i - x_j}, \quad j = 0, \ldots, i-1, \quad l_{ii}^{(-1)} = \frac{1}{\prod_{j=0}^{i-1} (x_i - x_j)}, \quad i = 0, \ldots, n. \tag{13}$$

In order to get the inverse of the matrix U, we start with the relation between the Newton basis and the monomial basis

$$(\omega_0(x), \ldots, \omega_n(x)) = (m_0(x), \ldots, m_n(x))U^{-1}.$$

Taking into account that the entries $u_{ij}^{(-1)}, i = 0, \ldots, j$, are the coefficients of ω_j with respect to the monomial basis $(m_0, \ldots, m_j)$, we can compare the coefficients in the relation $\omega_j(x) = (x - x_{j-1})\omega_{j-1}(x)$ and obtain the algorithm

$$\begin{aligned} &u_{00}^{(-1)} = 1, \quad u_{0j}^{(-1)} = -x_{j-1}u_{0,j-1}^{(-1)}, \quad j = 1, \ldots, n, \\ &u_{ii}^{(-1)} = 1, \quad u_{ij}^{(-1)} = u_{i-1,j-1}^{(-1)} - x_{j-1}u_{i,j-1}^{(-1)}, \quad j = i+1, \ldots, n, \quad i = 1, \ldots, n. \end{aligned} \tag{14}$$

§5. Accurate LU factorization of a Vandermonde matrix

A quantity $\mathbf{X}$ can be obtained with *high relative accuracy* (HRA) if the relative error of the computed value $\widehat{\mathbf{X}}$ can be bounded as follows:

$$\frac{||\mathbf{X} - \widehat{\mathbf{X}}||}{||\mathbf{X}||} \leq Cu,$$

where C is a positive constant independent of the arithmetic precision and u is the unit round-off.

In [2], it was shown we can compute with high relative accuracy the products, quotients and true additions (addition of numbers with the same sign) of expressions that can be computed with high relative accuracy. The subtractions (addition of numbers with the opposite sign) are permitted only with initial data of the problem, as shown in [2]. If we perform the computations with high relative accuracy, we can ensure that the relative errors are of the order of the roundoff unit, with independence of the conditioning of the problem.

It is well-known that the inverse of a Vandermonde matrix with nonnegative increasing nodes can be computed with HRA because the matrix is totally nonnegative and its bidiagonal factorization can be obtain with high relative accuracy (see [3] and [7]). However, the following result shows that such computation is possible whenever all distinct nodes have the same nonstrict sign and for all possible order configurations of the nodes.

Theorem 1. *Let V be the Vandermonde matrix in (1) corresponding to a sequence of distinct nodes of the same nonstrict sign. Then, V^{-1} can be computed with HRA.*

Proof. Formula (4) expresses the entries of V^{-1} as a quotient. The denominator is a product of differences of nodes (the initial data). The numerator is up to a sign a sum of terms. Such terms are products of nodes and have the same sign. □

We have the following results about the matrices L and U and their inverses. For the matrices L and L^{-1} there are no restrictions on the nodes, that is, HRA can be ensured for all posible signs and order configurations of the nodes.

Theorem 2. *Let L be the lower triangular matrix in (6). Then L and L^{-1} can be computed with HRA for any distinct nodes, x_i, $i = 0, \ldots, n$, using algorithms (8) and (13), respectively.*

Proof. The algorithms (8) and (13) only use subtractions of nodes (the initial data) and products and quotients of them. □

In contrast to Theorem 2, the following result requires the same restrictions on the nodes as in Theorem 1 to ensure that U and U^{-1} can be computed with HRA.

Theorem 3. *Let U be the upper triangular matrix in (10) corresponding to a sequence of nodes with the same nonstrict sign. Then U and U^{-1} can be computed with HRA, using algorithms (11) and (14), respectively.*

Proof. Let us first assume that all nodes are nonnegative, that is, $x_i \geq 0$ for all $i = 0, \ldots, n$.

According to (11), the first row and the diagonal have nonnegative entries that can be computed using only products of initial data. The elements u_{ij}, $j = i + 1, \ldots, n$, $i = 1, \ldots, n$, are computed by adding two nonnegative terms. Thus HRA can be ensured for the entries of U.

Analogously, we see from (14) that the (i, j) entries in the first row and the diagonal of U^{-1} have the same sign as $(-1)^{i+j}$ and can be computed with HRA. Then we see that the elements $u_{ij}^{(-1)}$, $j = i + 1, \ldots, n$, $i = 1, \ldots, n$, are the sum of two terms that can be computed with HRA, both of them with the same sign as $(-1)^{i+j}$.

Now let us assume that all nodes are nonpositive. Then, we have that

$$V(x_0, \ldots, x_n) = V(-x_0, \ldots, -x_n)S, \quad S := \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & \ddots & \\ & & & (-1)^n \end{pmatrix},$$

that is, the Vandermonde matrix coincides with a Vandermonde matrix with nonnegative nodes up to a change of sign in the columns. Let LU and $L_2 U_2$ be the Crout factorizations of $V(x_0, \ldots, x_n)$ and $V(-x_0, \ldots, -x_n)$, respectively. Taking into account $LU = (L_2 S)(S U_2 S)$, we deduce from the uniqueness of the Crout factorization that

$$L = L_2 S, \quad U = S U_2 S. \tag{15}$$

Therefore, $(-1)^{i+j} u_{ij}$ and $u_{ij}^{(-1)}$ are nonnegative. Now the proof follows analogously to the case of nonnegative nodes taking into account that the additions in algorithm (11) and (14) sum terms of the same sign. □

Acknowledgements

This work has been partially supported by the Spanish Research Grant MTM2012-31544, by Gobierno the Aragón and Fondo Social Europeo.

References

[1] de Boor, C., and Pinkus, A. Backward error analysis for totally positive linear systems. *Numer. Math 27* (1976/1977), 485–490.

[2] Demmel, J., Gu, M., Eisenstat, S., Slapničar, I., Veselić, K., and Drmač, Z. Computing the singular value decomposition with high relative accuracy. *Lineal Algebra and its Applications 299* (1999), 21–80.

[3] Demmel, J., and Koev, P. The accurate and efficient solution of a totally positive generalized vandermonde linear system. *SIAM J. Matrix Anal. Appl. 27* (2005), 142–152.

[4] Gasca, M., and Peña, J. M. On factorizations of totally positive matrices. In *Total positivity and its applications*, M. Gasca and C. A. Micchelli, Eds., vol. 359 of *Math. Appl.* Kluwer Acad. Publ., Dordrecht, 1994, pp. 109–130.

[5] Higham, N. J. Stability analysis of algorithms for solving confluent vandermonde-like systems. *SIAM J. Matrix Anal. Appl. 1* (1990), 23–41.

[6] Higham, N. J. *Accuracy and Stability of Numerical Algorithms*. Siam, Philadelphia, 1996.

[7] Koev, P. Accurate computations with totally nonnegative matrices. *SIAM J. Matrix Anal. Appl. 29* (2007), 731–751.

[8] Peña, J. M. Pivoting strategies leading to small bounds of the errors for certain linear systems. *IMA Journal of Numerical Analysis 16* (1996), 141–153.

[9] Reichel, L. Newton interpolation at leja points. *BIT 30*, 2 (1990), 332–346.

J. Carnicer, Y. Khiar and J. M. Peña
Departamento de Matemática Aplicada
Universidad de Zaragoza
C/Pedro Cerbuna, 12, 50009 Zaragoza
carnicer@unizar.es, yasmina@unizar.es and jmpena@unizar.es