Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 924757, 7 pages
http://dx.doi.org/10.1155/2015/924757

Research Article

A New Derivation and Recursive Algorithm Based on Wronskian Matrix for Vandermonde Inverse Matrix

Qun Zhou,[1] Xinjian Zhang,[2] and Xiongwei Liu[2]

[1] *School of Information Science and Engineering, Hunan International Economics University, Changsha, Hunan 410205, China*
[2] *College of Science, National University of Defense Technology, Changsha, Hunan 410073, China*

Correspondence should be addressed to Xinjian Zhang; xjz_20075@163.com

Received 26 June 2015; Accepted 27 September 2015

Academic Editor: Xinkai Chen

For an analytical expression of Vandermonde inverse matrix, a new derivation process based on Wronskian matrix and Lagrange interpolation polynomial basis is presented. Recursive formula and implementation cases for the direct formula of Vandermonde inverse matrix are given based on deriving the unified formula of Wronskian inverse matrix. For the calculation of symbol-type Vandermonde inverse matrix, the direct formula and recursive method are verified to be more efficient than Mathematica which is good at symbolic computation by comparing the computing time in Mathematica. The process and steps of recursive algorithm are relatively simple. The derivation process and idea both have very important values in theory and practice of Vandermonde and generalized Vandermonde inverse matrix.

1. Introduction

Vandermonde matrix and generalized Vandermonde matrix and their inverses are always widely concerned in many research fields, such as numerical analysis, data and signal processing, and control theory [1–8]. There are many effective ways to calculate the Vandermonde inverse matrix. Fox example, Tou [1] and Reis [2] obtained the matrix formula from the coefficients of polynomial terms. Neagoe [3] deduced an analytic formula of complex-type Vandermonde inverse matrix based on symmetric polynomials. Eisinberg and Fedele [4] presented a general explicit formula for the elements of inverse matrix and two different algorithms were deduced. In this paper, based on results in [1, 2], a new derivation process of an analytical expression of Vandermonde inverse matrix is presented based on Wronskian matrix and Lagrange interpolation polynomial basis, and recursive formula and implementation cases for the direct formula of Vandermonde inverse matrix are presented based on deducing the unified formula of Wronskian inverse matrix.

The rest of this paper is arranged as follows. Firstly, we summarize the main idea and analytical formulas presented in [3] and give a new derivation process for the formula in Section 2. In Section 3, a recursive formula of Vandermonde inverse matrix is deduced. In Section 4, numerical simulations are presented with some cases in Mathematica and conclusion and prospect are given by comparing and analyzing the numerical results in the last section.

2. A New Derivation of Vandermonde Inverse Matrix

If $x_1, x_2, \ldots, x_n$ $(n \geq 1)$ are different numbers, that is to say, they are not equal to each other, then we define the n-order matrix

$$V_n(x_1, x_2, \ldots, x_n) = (v_{ij})_{n\times n} = (x_j^{i-1})_{n\times n} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{pmatrix}, \quad (1)$$

where $i, j = 1, 2, \ldots, n$ as a Vandermonde matrix. For the inverse of this matrix, Neagoe [3] obtained the calculation formula (2) according to relations between the determinant of the matrix (1) and the determinant of the matrix which was obtained by adding a row whose elements were x_j^n ($j = 1, 2, \ldots, n$) and a column whose elements were x^i ($i = 0, 1, 2, \ldots, n$) to (1) and based on the calculation formula of inverse matrix by using the matrix determinant:

$$V_n^{-1} = \left[\left(v_{ij}^{-1}\right)_{n\times n}\right]^T = \left[\left((-1)^{i+j} \cdot \frac{\sigma_{n-i}(j)}{\prod_{k=1}^{j-1}\left(x_j - x_k\right)\prod_{k=j+1}^{n}\left(x_k - x_j\right)}\right)_{n\times n}\right]^T, \tag{2}$$

where $i = 1, 2, \ldots, n$ are column subscripts and $j = 1, 2, \ldots, n$ are row subscripts and

$$\begin{gathered}\sigma_0\left(x_1, x_2, \ldots, x_n\right) = 1, \\ \sigma_k\left(x_1, x_2, \ldots, x_n\right) = \sum_{1\le j_1<\cdots<j_k\le n} x_{j_1}x_{j_2}\cdots x_{j_k}, \\ k = 1, 2, \ldots, n, \\ \sigma_k(j) = \sigma_k\left(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n\right).\end{gathered} \tag{3}$$

Here we will use a new method entirely different from [3] to deduce formula (2).

For a group of linearly independent functions $\Phi = \{\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x)\}$, where $\varphi_i(x)$ ($i = 1, 2, \ldots, n$) has the $(n-1)$th derivative, Wronskian matrix is defined as

$$\begin{gathered}W\left[\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x)\right] = \left(w_{ij}\right)_{n\times n} = \left(\varphi_j^{(i)}\right)_{n\times n}, \\ j = 1, 2, \ldots, n;\ i = 0, 1, \ldots, n-1.\end{gathered} \tag{4}$$

If there is another group of functions $\widetilde{\Phi} = \{\widetilde{\varphi}_1(x), \widetilde{\varphi}_2(x), \ldots, \widetilde{\varphi}_n(x)\}$ which satisfy

$$\sum_{i=1}^{n}\varphi_i(x)\,\widetilde{\varphi}_i(t) = \frac{(x-t)^{n-1}}{(n-1)!}, \tag{5}$$

then the inverse matrix $W^{-1}[\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x)]$ of (4) can be obtained [9] as

$$W^{-1}\left[\varphi_1(x), \ldots, \varphi_n(x)\right] = \left[(-1)^{n-j}\,\widetilde{\varphi}_i^{(n-j)}(x)\right]_{n\times n}, \tag{6}$$

where $j = 1, 2, \ldots, n$ and $i = 1, \ldots, n$.

If we set $y_i(x) = (1/(n-1)!)(x - x_i)^{n-1}$ ($i = 1, 2, \ldots, n$), then we can verify that the Lagrange interpolation polynomial basis

$$l_i(x) = \frac{1}{w'(x_i)}\prod_{\substack{j=1\\ j\neq i}}^{n}\left(x - x_j\right) \quad (i = 1, 2, \ldots, n), \tag{7}$$

where $w(x) = \prod_{i=1}^{n}(x - x_i)$ satisfy (5); that is to say,

$$\sum_{i=1}^{n} y_i(x)\,l_i(t) = \frac{(x-t)^{n-1}}{(n-1)!}. \tag{8}$$

For Wronskian matrix $W[y_1(x), \ldots, y_n(x)]$,

$$\begin{aligned}W\left[y_1(x), \ldots, y_n(x)\right] &= \left[\frac{1}{(n-i)!}\left(x - x_j\right)^{n-i}\right]_{n\times n} \\ &= \begin{bmatrix} \frac{1}{(n-1)!}\left(x - x_1\right)^{n-1} & \frac{1}{(n-1)!}\left(x - x_2\right)^{n-1} & \cdots & \frac{1}{(n-1)!}\left(x - x_n\right)^{n-1} \\ \frac{1}{(n-2)!}\left(x - x_1\right)^{n-2} & \frac{1}{(n-2)!}\left(x - x_2\right)^{n-2} & \cdots & \frac{1}{(n-2)!}\left(x - x_n\right)^{n-2} \\ \vdots & \vdots & \cdots & \vdots \\ x - x_1 & x - x_2 & \cdots & x - x_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}.\end{aligned} \tag{9}$$

From (6), the inverse matrix of $W[y_1(x),\dots,y_n(x)]$ is

$$W^{-1}\left[y_1(x),\dots,y_n(x)\right]=\left[(-1)^{n-j}\,l_i^{(n-j)}(x)\right]_{n\times n}$$
$$=\begin{bmatrix}(-1)^{n-1}\,l_1^{(n-1)}(x) & (-1)^{n-2}\,l_1^{(n-2)}(x) & \cdots & l_1(x)\\ (-1)^{n-1}\,l_2^{(n-1)}(x) & (-1)^{n-2}\,l_2^{(n-2)}(x) & & l_2(x)\\ \vdots & \vdots & \vdots & \vdots\\ (-1)^{n-1}\,l_n^{(n-1)}(x) & (-1)^{n-2}\,l_n^{(n-2)}(x) & & l_n(x)\end{bmatrix}, \qquad (10)$$

where $i=1,2,\dots,n$ are row subscripts and $j=1,2,\dots,n$ are column subscripts in the above two matrices.

Denote by D_n the diagonal matrix composed of $\{(-1)^{n-i}(n-i)!\}_{i=1,2,\dots,n-1,n}$; that is,

$$D_n=\begin{bmatrix}(-1)^{n-1}(n-1)! & 0 & \cdots & 0\\ 0 & (-1)^{n-2}(n-2)! & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1\end{bmatrix}. \qquad (11)$$

It is easily proved that

$$D_n\cdot W\left[y_1(x),\dots,y_n(x)\right]=\left[\left(x_j-x\right)^{n-i}\right]_{n\times n}$$
$$=\begin{bmatrix}(x_1-x)^{n-1} & (x_2-x)^{n-1} & \cdots & (x_n-x)^{n-1}\\ (x_1-x)^{n-2} & (x_2-x)^{n-2} & \cdots & (x_n-x)^{n-2}\\ \vdots & \vdots & \vdots & \vdots\\ x_1-x & x_2-x & \cdots & x_n-x\\ 1 & 1 & \cdots & 1\end{bmatrix}. \qquad (12)$$

Denoted by

$$A_n(x,x_1,\dots,x_n)=\left[\left(x_j-x\right)^{n-i}\right]_{n\times n}, \qquad (13)$$

using inverse matrix formula of matrix multiplication, we have

$$\begin{aligned}\left(D_n\cdot W\left[y_1(x),\dots,y_n(x)\right]\right)^{-1}&=W^{-1}\left[y_1(x),\dots,y_n(x)\right]\cdot D_n^{-1}\\&=A_n^{-1}(x,x_1,\dots,x_n).\end{aligned} \qquad (14)$$

From (10) and (14), we can get

$$\begin{aligned}A_n^{-1}(x,x_1,\dots,x_n)&=\left[\frac{(-1)^{n-j}}{(n-j)!}\cdot(-1)^{n-j}\,l_i^{(n-j)}(x)\right]_{n\times n}\\&=\left[\frac{l_i^{(n-j)}(x)}{(n-j)!}\right]_{n\times n}.\end{aligned} \qquad (15)$$

Let $x=0$; then (12) changes to

$$A_n(0,x_1,\dots,x_n)=\begin{bmatrix}x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1}\\ x_1^{n-2} & x_2^{n-2} & \cdots & x_n^{n-2}\\ \vdots & \vdots & \vdots & \vdots\\ x_1 & x_2 & \cdots & x_n\\ 1 & 1 & \cdots & 1\end{bmatrix}. \qquad (16)$$

Its inverse matrix is

$$A_n^{-1}(0,x_1,\dots,x_n)=\left[\frac{l_i^{(n-j)}(0)}{(n-j)!}\right]_{n\times n}. \qquad (17)$$

For the values of $\sigma_k(x_1,\dots,x_n)$ $(k=1,2,\dots,n)$ in (2), we specify them by the following case. For example, taking $n=3$, from (3), we can get

$$\begin{aligned}\sigma_0(x_1,x_2,x_3)&=1,\\ \sigma_1(x_1,x_2,x_3)&=x_1+x_2+x_3,\\ \sigma_2(x_1,x_2,x_3)&=x_1x_2+x_1x_3+x_2x_3,\\ \sigma_3(x_1,x_2,x_3)&=x_1x_2x_3.\end{aligned} \qquad (18)$$

Then we have

$$l_i^{(n-j)}(0)=(-1)^{i+j}\frac{(n-j)!\sigma_{j-1}(i)}{\prod_{k=1}^{i-1}(x_i-x_k)\cdot\prod_{k=i+1}^{n}(x_k-x_i)}, \qquad (19)$$

where $i=1,2,\dots,n$ are row subscripts and $j=1,2,\dots,n$ are column subscripts, and from (17) and (19), we can get

$$\begin{aligned}&A_n^{-1}(0,x_1,\dots,x_n)\\&\quad=\left[(-1)^{i+j}\frac{\sigma_{j-1}(i)}{\prod_{k=1}^{i-1}(x_i-x_k)\cdot\prod_{k=i+1}^{n}(x_k-x_i)}\right]_{n\times n}.\end{aligned} \qquad (20)$$

Thus, according to the corresponding relations of elements between $A_n(0,x_1,\dots,x_n)$ and $V_n(x_1,\dots,x_n)$ and $A_n^{-1}(0,x_1,\dots,x_n)$ and $V_n^{-1}(x_1,\dots,x_n)$, we can obtain (2); that is,

$$\begin{aligned}&V_n^{-1}(x_1,\dots,x_n)\\&\quad=\left[\frac{(-1)^{i+j}\sigma_{n-j}(i)}{\prod_{k=1}^{i-1}(x_i-x_k)\cdot\prod_{k=i+1}^{n}(x_k-x_i)}\right]_{n\times n},\end{aligned} \qquad (21)$$

where $i=1,2,\dots,n$ are row subscripts and $j=1,2,\dots,n$ are column subscripts.

3. Recursive Algorithm of Vandermonde Inverse Matrix

By the above method of function matrix, we denote $V_n(x, x_1, \dots, x_n)$ as

$$V_n(x, x_1, \dots, x_n) = \left[\left(x_j - x\right)^{i-1}\right]_{n\times n} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 - x & x_2 - x & \cdots & x_m - x \\ (x_1 - x)^2 & (x_2 - x)^2 & \cdots & (x_m - x)^2 \\ \vdots & \vdots & \vdots & \vdots \\ (x_1 - x)^{n-1} & (x_2 - x)^{n-1} & \cdots & (x_m - x)^{n-1} \end{pmatrix}. \quad (22)$$

According to the property of Vandermonde determinant, we have

$$\begin{aligned} \det V_n(x, x_1, \dots, x_n) &= \prod_{1\le i<j\le n} \left[\left(x_j - x\right) - \left(x_i - x\right)\right] \\ &= \prod_{1\le i<j\le n} \left(x_j - x_i\right). \end{aligned} \quad (23)$$

For the convenience of description, we introduce the following notations.

$A_n(i, j, x)$ denotes the algebraic complement of the element of $V_n(x, x_1, \dots, x_n)$ whose row subscript is i and column subscript is j. $A_n^{(k)}(i, j, x)$ denotes kth derivative on x of $A_n(i, j, x)$ and

$$\begin{aligned} B_{nj}(x) &= \prod_{\substack{1\le k\le n \\ k\ne j}} \left(x_k - x\right), \\ C_{nj} &= \prod_{\substack{1\le i<k\le n \\ i,k\ne j}} \left(x_k - x_i\right), \\ & \qquad j = 1, 2, \dots, n. \end{aligned} \quad (24)$$

It is easily obtained that

$$A_n^{(1)}\left(i-1, j, x\right) = (i-1)\, A_n\left(i, j, x\right), \quad (25)$$

$$\begin{aligned} B_{nj}(x) &= \left(x_n - x\right) B_{n-1,j}(x), \\ B_{nn}(x) &= \left(x_j - x\right) B_{n-1,j}(x) \\ & \qquad \left(j = 1, 2, \dots, n-1\right). \end{aligned} \quad (26)$$

From (25) and (26), we can get

$$\begin{aligned} A_n\left(i, j, x\right) &= \frac{1}{(i-1)!} A_n^{(i-1)}\left(1, j, x\right) \\ & \qquad (i = 1, 2, \dots, n), \\ A_n\left(1, j, x\right) &= (-1)^{j+1} C_{nj} B_{nj}(x), \quad 1 \le j \le n, \\ A_n\left(i, j, x\right) &= \frac{(-1)^{j+1}}{(i-1)!} C_{nj} B_{nj}^{(i-1)}(x), \quad 1 \le i, \ j \le n. \end{aligned} \quad (27)$$

Calculating kth derivative on x at both sides of (26), we can obtain

$$\begin{aligned} \left[B_{nj}(x)\right]^{(k)} &= \left(x_n - x\right) B_{n-1,j}^{(k)}(x) - kB_{n-1,j}^{(k-1)}(x), \\ \left[B_{nn}(x)\right]^{(k)} &= \left(x_j - x\right) B_{n-1,j}^{(k)}(x) - kB_{n-1,j}^{(k-1)}(x), \\ & \qquad j = 1, 2, \dots, n-1. \end{aligned} \quad (28)$$

They can lead to

$$\begin{aligned} A_n\left(i, j, x\right) &= \frac{(-1)^{j+1} C_{nj}}{(i-1)!} \left[\left(x_n - x\right) B_{n-1,j}^{(i-1)}(x) \right. \\ & \quad \left. - (i-1)\, B_{n-1,j}^{(i-2)}(x)\right], \quad 1 \le i \le n, \ 1 \le j \le n-1, \\ A_n\left(i, n, x\right) &= \frac{(-1)^{n+1} C_{nn}}{(i-1)!} \left[\left(x_{n-1} - x\right) B_{n-1,n-1}^{(i-1)}(x) \right. \\ & \quad \left. - (i-1)\, B_{n-1,n-1}^{(i-2)}(x)\right], \quad 1 \le i \le n. \end{aligned} \quad (29)$$

It is easy to know that

$$\begin{aligned} C_{nj} &= \prod_{\substack{1\le k\le n-1 \\ k\ne j}} \left(x_n - x_k\right) \cdot C_{n-1,j} \\ & \qquad \left(1 \le j \le n-1\right), \\ C_{nn} &= \prod_{1\le k\le n-2} \left(x_{n-1} - x_k\right) \cdot C_{n-1,n-1}, \\ \det V_n(t) &= \prod_{1\le k\le n-1} \left(x_n - x_k\right) \det V_{n-1}(x). \end{aligned} \quad (30)$$

Denoted by $V_n^{-1}(x, x_1, \dots, x_n) = [\tilde{v}_n(i, j, x)]_{n\times n}$, then

$$\tilde{v}_n\left(i, j, x\right) = \frac{A_n\left(j, i, x\right)}{\det V_n(x)}. \quad (31)$$

From (29)–(31), we can get

$$\begin{aligned} \tilde{v}_n\left(i, j, x\right) &= \frac{1}{x_n - x_i} \left[\left(x_n - x\right) \tilde{v}_{n-1}\left(i, j, x\right) \right. \\ & \quad \left. - \tilde{v}_{n-1}\left(i, j-1, x\right)\right], \quad 1 \le i \le n-1, \ 1 \le j \le n, \\ \tilde{v}_n\left(n, j, x\right) &= p_n \left[\left(x_{n-1} - x\right) \tilde{v}_{n-1}\left(n-1, j, x\right) \right. \\ & \quad \left. - \tilde{v}_{n-1}\left(n-1, j-1, x\right)\right], \quad 1 \le j \le n. \end{aligned} \quad (32)$$

And set $\tilde{v}_n(i, j, x) = 0$ when $i \le 0$, $i \le 0$, $j \le 0$, $i > n$, and $j > n$.

Let $x = 0$ in (32); the recursion formula of elements of Vandermonde inverse matrix $V_n^{-1}(0, x_1, \dots, x_n) = (\tilde{v}_n(i, j, 0))_{n\times n}$ can be obtained.

For simplicity, denote $\tilde{v}_n(i, j, 0) = \tilde{v}_n(i, j)$. Because $x = 0$, all elements do not contain x, so we denote n-order Vandermonde matrix composed of $x_1, \dots, x_n$ as

$$\begin{aligned} V_n^{-1}\left(x_1, \dots, x_n\right) &= \left(v_n\left(i, j\right)\right)_{n\times n} \\ &= \left[(-1)^{i+1}\, \tilde{v}_n\left(i, j\right)\right]_{n\times n}. \end{aligned} \quad (33)$$

Thus, each element of Vandermonde inverse matrix can be determined by the following recursive formula:

$$\widetilde{v}_n(i,j)=\begin{cases}\dfrac{1}{x_n-x_i}\left[x_n\widetilde{v}_{n-1}(i,j)-\widetilde{v}_{n-1}(i,j-1)\right], & 1\le i\le n-1,\ 1\le j\le n,\\[2ex] p_n\left[x_{n-1}\widetilde{v}_{n-1}(n-1,j)-\widetilde{v}_{n-1}(n-1,j-1)\right], & 1\le j\le n,\end{cases}\tag{34}$$

where $p_n=\prod_{1\le k\le n-2}(x_{n-1}-x_k)/\prod_{1\le k\le n-1}(x_n-x_k)$ and set $\widetilde{v}_n(i,j)=0$ when $i\le 0$, $i\le 0$, $j\le 0$, $i>n$, and $j>n$.

When $n=1$, $\prod_{1\le k\le n-1}(x_n-x_k)=1$ and $\prod_{1\le k\le n-2}(x_n-x_k)=1$.

When $n=2$, $\prod_{1\le k\le n-2}(x_n-x_k)=1$.

4. Application Case and Simulation

We will take the 3-order Vandermonde inverse matrix as an example to illustrate the recursive steps of (34) in detail.

Let

$$V_3(x_1,x_2,x_3)=\begin{pmatrix}1&1&1\\x_1&x_2&x_3\\x_1^2&x_2^2&x_3^2\end{pmatrix}.\tag{35}$$

From (34), we have

$$\begin{aligned}V_3^{-1}(x_1,x_2,x_3)&=(v_3(i,j))_{3\times3}\\&=\left[(-1)^{i+1}\widetilde{v}_3(i,j)\right]_{3\times3}.\end{aligned}\tag{36}$$

Taking $\widetilde{v}_1(1,1)=1$ and $p_1=1$ as the initial values to calculate each element of $[\widetilde{v}_3(i,j)]_{3\times3}$, then we can get

$$\widetilde{v}_3(1,1)=\frac{1}{x_3-x_1}\left[x_3\widetilde{v}_2(1,1)-\widetilde{v}_2(1,0)\right].\tag{37}$$

Since $\widetilde{v}_2(1,0)=0$ $(j=0\le 0)$, then

$$\widetilde{v}_3(1,1)=\frac{x_3}{x_3-x_1}\widetilde{v}_2(1,1).\tag{38}$$

From (34) again, we can get

$$\widetilde{v}_2(1,1)=\frac{1}{x_2-x_1}\left[x_2\widetilde{v}_1(1,1)-\widetilde{v}_1(1,0)\right].\tag{39}$$

Since $\widetilde{v}_1(1,0)=0$ $(j=0\le 0)$, then

$$\widetilde{v}_2(1,1)=\frac{x_2}{x_2-x_1}\widetilde{v}_1(1,1).\tag{40}$$

Substituting $\widetilde{v}_1(1,1)=1$ to (40), we have

$$\begin{aligned}\widetilde{v}_3(1,1)&=\frac{x_3}{x_3-x_1}\widetilde{v}_2(1,1)=\frac{x_2}{x_2-x_1}\cdot\frac{x_3}{x_3-x_1}\\&=\frac{x_2x_3}{(x_2-x_1)(x_3-x_1)}.\end{aligned}\tag{41}$$

As $i=1$ in $\widetilde{v}_3(1,1)$, so from (36), this leads to

$$v_3(1,1)=(-1)^{1+1}\widetilde{v}_3(1,1)=\frac{x_2x_3}{(x_2-x_1)(x_3-x_1)}.\tag{42}$$

Meanwhile, we can also obtain

$$v_2(1,1)=(-1)^{1+1}\widetilde{v}_2(1,1)=\frac{x_2}{x_2-x_1}.\tag{43}$$

For $\widetilde{v}_3(2,3)$, there is

$$\widetilde{v}_3(2,3)=\frac{1}{x_3-x_2}\left[x_3\widetilde{v}_2(2,3)-\widetilde{v}_2(2,2)\right].\tag{44}$$

As $j=3>n=2$ in $\widetilde{v}_2(2,3)$, so $\widetilde{v}_2(2,3)=0$, and from (35), we can obtain

$$\begin{aligned}\widetilde{v}_3(2,3)&=-\frac{1}{x_3-x_2}\widetilde{v}_2(2,2)\\&=-\frac{1}{x_3-x_2}p_2\left[x_1\widetilde{v}_1(1,2)-\widetilde{v}_1(1,1)\right].\end{aligned}\tag{45}$$

As $j=2>n=1$ in $\widetilde{v}_1(1,2)$, so $\widetilde{v}_1(1,2)=0$ and $\widetilde{v}_1(1,1)=1$ and

$$p_2=\frac{\prod_{1\le k\le 0}(x_1-x_k)}{\prod_{1\le k\le 1}(x_2-x_k)}=\frac{1}{x_2-x_1}.\tag{46}$$

From (45) and (36), $i=2$; then we can get $v_3(2,3)$; that is,

$$\begin{aligned}v_3(2,3)&=(-1)^{2+1}\widetilde{v}_3(2,3)=(-1)^{2+1}\frac{-1}{x_3-x_2}\frac{-1}{x_2-x_1}\\&=-\frac{1}{(x_1-x_2)(x_2-x_3)}.\end{aligned}\tag{47}$$

Meanwhile, we can also get

$$\begin{aligned}v_2(2,2)&=(-1)^{2+1}\widetilde{v}_2(2,2)=(-1)^{2+1}\left[-p_2\widetilde{v}_1(1,1)\right]\\&=p_2=\frac{1}{x_2-x_1}.\end{aligned}\tag{48}$$

The other elements of $\widetilde{v}_3(i,j)$, $1\le i,j\le 3$, can be obtained by a similar process. Finally, we can get

Table 1: The total time spent on symbolic Vandermonde inverse matrix.

n	3	4	5	6	7	8	9	10
MMA	0.0156	0.1248	0.8268	2.4648	38.5164	282.36	2549.04	26001
Equation (2)	0.0624	0.1092	0.2184	0.4212	0.7800	1.4508	2.8236	5.6940
Equation (34)	0.1560	0.3120	0.9360	2.4960	6.2400	15.4440	36.0360	82.3681

Table 2: The total time spent on numerical Vandermonde inverse matrix.

n	3	4	5	6	7	8	9	10
MMA	0	0	0	0.0156	0.0156	0.0312	0.0468	0.0468
Equation (2)	0.0468	0.0936	0.1560	0.2652	0.4992	0.9204	1.7784	2.6364
Equation (34)	0.0624	0.2340	0.6552	1.7628	4.3836	10.6392	24.4920	55.8481

$$\left(\tilde{v}_3\left(i,j\right)\right)_{3\times 3}=\begin{pmatrix} \dfrac{x_2x_3}{\left(x_1-x_2\right)\left(x_1-x_3\right)} & -\dfrac{x_2+x_3}{\left(x_1-x_2\right)\left(x_1-x_3\right)} & \dfrac{1}{\left(x_1-x_2\right)\left(x_1-x_3\right)} \\ \dfrac{x_1x_3}{\left(x_1-x_2\right)\left(x_2-x_3\right)} & -\dfrac{x_1+x_3}{\left(x_1-x_2\right)\left(x_2-x_3\right)} & \dfrac{1}{\left(x_1-x_2\right)\left(x_2-x_3\right)} \\ \dfrac{x_1x_2}{\left(x_3-x_1\right)\left(x_3-x_2\right)} & -\dfrac{x_1+x_2}{\left(x_3-x_1\right)\left(x_3-x_2\right)} & \dfrac{1}{\left(x_3-x_1\right)\left(x_3-x_2\right)} \end{pmatrix}. \qquad (49)$$

Meanwhile, we can get

$$\left(\tilde{v}_2\left(i,j\right)\right)_{2\times 2}=\begin{pmatrix} \dfrac{x_2}{x_2-x_1} & \dfrac{1}{x_1-x_2} \\ \dfrac{x_1}{x_2-x_1} & \dfrac{1}{x_1-x_2} \end{pmatrix}. \qquad (50)$$

So, the 3-order Vandermonde inverse matrix of (35) is

$$V_2^{-1}\left(x_1,x_2,x_3\right)=\left(\left(-1\right)^{i+1}\tilde{v}_3\left(i,j,0\right)\right)_{3\times 3}=\begin{pmatrix} \dfrac{x_2x_3}{\left(x_1-x_2\right)\left(x_1-x_3\right)} & -\dfrac{x_2+x_3}{\left(x_1-x_2\right)\left(x_1-x_3\right)} & \dfrac{1}{\left(x_1-x_2\right)\left(x_1-x_3\right)} \\ -\dfrac{x_1x_3}{\left(x_1-x_2\right)\left(x_2-x_3\right)} & \dfrac{x_1+x_3}{\left(x_1-x_2\right)\left(x_2-x_3\right)} & -\dfrac{1}{\left(x_1-x_2\right)\left(x_2-x_3\right)} \\ \dfrac{x_1x_2}{\left(x_3-x_1\right)\left(x_3-x_2\right)} & -\dfrac{x_1+x_2}{\left(x_3-x_1\right)\left(x_3-x_2\right)} & \dfrac{1}{\left(x_3-x_1\right)\left(x_3-x_2\right)} \end{pmatrix} \qquad (51)$$

and the 2-order Vandermonde inverse matrix is

$$\begin{aligned}\left(v_2\left(i,j\right)\right)_{2\times 2}&=\left(\left(-1\right)^{i+1}v_2\left(i,j\right)\right)_{2\times 2}\\ &=\begin{pmatrix} \dfrac{x_2}{x_2-x_1} & \dfrac{1}{x_1-x_2} \\ \dfrac{x_1}{x_1-x_2} & \dfrac{1}{x_2-x_1} \end{pmatrix}.\end{aligned} \qquad (52)$$

We choose different order Vandermonde matrices to compare the computing time spent on calculating their inverse matrices in mathematical software Mathematica 10. We divide the calculations into two types: symbolic inverse matrix and numerical inverse matrix. For the limitation of computing time, we take only the order as $n = 3,\ldots,10$ and the total time of each calculation is the time spent by doing 100 repeated calculations; its unit is second. When we calculate the numerical Vandermonde inverse matrix, we take $x_i = i$ $(i = 1,\ldots,n)$. We obtain the results as Tables 1 and 2.

In Tables 1 and 2, n is the order of Vandermonde inverse matrix. MMA denotes the time which Mathematica spends on calculation by using its inner function "Inverse[ ]" to calculate the inverse matrix. Equations (2) and (34) denote the times which Mathematica spends on calculations by (2) and (34) separately.

Remark 1. Here we only take Mathematica as the execution environment, mainly because Mathematica has the superiority of symbolic computation compared with other types of commonly used mathematical software. For example, when we calculate the 6-order symbolic Vandermonde inverse matrix by the inner functions of Mathematica, Maple, and MATLAB, the computing times are 0.187, 0.281, and 22.013 seconds separately.

5. Conclusion and Prospect

From Table 1, we can see that the total time spent by Mathematica on calculating the symbolic Vandermonde inverse matrix by using inner function "Inverse[ ]" is far more than that by (2) and (34). With the increase of dimensions, the time grows dramatically. For the 10-order matrix, it almost spends nearly 7 hours. From Table 2, we can see that the total time spent by Mathematica (0 in Table 2 means that the total time is less than 0.001 seconds) on calculating the numerical Vandermonde inverse matrix by using inner function "Inverse[ ]" is far less than that by (2) and (34). For numerical inverse matrix, Mathematica's inner function has greater advantage than (2) and (34), and the time spent by (2) is less than that by (34). This is because our algorithms are not optimized in terms of numerical calculation. We do these calculations only by a few functions written by us in Mathematica. Therefore, the total computing times are much more. But for symbolic inverse matrix, (2) and (34) are better than Mathematica inner function.

In this paper we use the different methods from [2–5] to deduce (2) based on Wronskian function matrix and propose a recursive algorithm for Vandermonde inverse matrix. The process has a certain universality and some values for calculating inverse matrices of other special matrices. From Tables 1 and 2, we can see that two methods in this paper are more efficient than Mathematica inner function for symbolic Vandermonde inverse matrix. For numerical Vandermonde inverse matrix, how to optimize and design the effective algorithms to carry out our methods will be our future research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors thank the unknown reviewers for their careful reading and helpful comments. This work is supported by Science Research Project of Hunan Province Education Office (14C0650) and Research Project of National University of Defense Technology (JC12-02-01).

References

[1] J. T. Tou, "Determination of the inverse Vandermonde matrix," *IEEE Transactions on Automatic Control*, vol. 9, no. 3, p. 314, 1964.

[2] G. C. Reis, A Matrix Formulation for the Inverse Vandermonde Matrix, 1967.

[3] V.-E. Neagoe, "Inversion of the Van der Monde matrix," *IEEE Signal Processing Letters*, vol. 3, no. 4, pp. 119–120, 1996.

[4] A. Eisinberg and G. Fedele, "On the inversion of the Vandermonde matrix," *Applied Mathematics and Computation*, vol. 174, no. 2, pp. 1384–1397, 2006.

[5] X. F. Lu, Z. Xu, Q. Lu et al., "A fast algorithm of the inverse of Vandermonde-type matrix," in *Proceedings of the 7th International Conference on Matrix Theory and Its Application*, vol. 17, pp. 246–249, Chengdu, China, 2006.

[6] Z. Xu, *Fast Algorithm of Vandermonde Matrix Class*, Northwestern Polytechnic University Press, Xi'an, China, 1997.

[7] L. Zhao, Z. Xu, and Q. Lu, "An efficient and fast algorithm for the inverse of generalized Vandermonde matrix," *Numerical Mathematics: A Journal of Chinese Universities*, vol. 33, no. 4, pp. 312–318, 2011.

[8] P. R. Mane, S. G. Adiga, and M. Sathish Kumar, "Performance evaluation of random linear network coding using a Vandermonde matrix," *Physical Communication*, vol. 10, pp. 24–30, 2014.

[9] X. Zhang, *Theory and Application of Reproducing Kernel*, Science Press, Beijing, China, 2010.