On the inversion of the Vandermonde matrix

A. Eisinberg, G. Fedele *

Dip. Elettronica Informatica e Sistemistica, Università degli Studi della Calabria, 87036 Rende (Cs), Italy

Abstract

The inversion of the Vandermonde matrix has received much attention for its role in the solution of some problems of numerical analysis and control theory. This work deals with the problem of getting an explicit formula for the generic element of the inverse. We derive two algorithms in $O(n^2)$ and $O(n^3)$ and compare them with the Parker–Traub and the Björck–Pereyra algorithms.

Keywords: Vandermonde matrices; Parker–Traub algorithm; Björck–Pereyra algorithm

1. Introduction

We consider the inversion of the Vandermonde matrix,

$$ V(x_1, x_2, \ldots, x_n) = \begin{bmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ 1 & x_2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^{n-1} \end{bmatrix} $$

* Corresponding author.
E-mail address: fedele@si.deis.unical.it (G. Fedele).

© 2005 Elsevier Inc. All rights reserved.

doi:10.1016/j.amc.2005.06.014
where $x_i \in C^n$, $i = 1, 2, \ldots, n$. Vandermonde matrices play an important role both in mathematics and in applied sciences. The interest in this topic traces back to the sixties [1–4] and its relevance in the field of numerical analysis and signal processing is witnessed by the most recent contributions [5–7]. The most important applications of the Vandermonde matrix are probably in interpolation, a research field for which Erik Meijering, in his “Encouraged Paper” [8] writes:

“A search in the multidisciplinary databases of bibliographic information collected by the Institute for Scientific Information in the Web of Science will reveal that the number for publications containing the word ‘interpolation’ in the title, list of keywords, or the abstract has dramatically increased over the past decade, even when taking into account the intrinsic (and likewise dramatic) increase in the number of publications as a function of time."

Explicit formulas for solving Vandermonde systems and computing the inverse of the Vandermonde matrix are well known [1,2,4,9]; their relevance is due to the fact that Vandermonde problems are usually ill-conditioned and standard numerically stable methods in general fail to accurately compute the entries of the inverse or of the solution vector [10,11]. Therefore it is important to exploit the special structure of the Vandermonde matrix in order to derive algorithms both accurate and fast. Gohberg and Olshevsky, in their paper [12], said that to the best of their knowledge the possibility of simultaneously fast and accurate inversion of the Vandermonde matrix was not reported anywhere and proposed an accurate generalized Parker–Traub algorithm for the inverse of the Vandermonde matrix in $O(n^2)$. Moreover they used the well-known Björck and Pereyra algorithm [3] to invert the Vandermonde matrix in $O(n^3)$. In this paper we propose an explicit formula for the inverse which is a generalization of [13]. By using this formula we obtain two algorithms in $O(n^2)$ and $O(n^3)$. These algorithms seem to be competitive with the existing ones in terms of numerical accuracy and computational effort.

2. Proposed inversion formula

Let $X_n = t\{x_1, x_2, \ldots, x_n\}$ be a set of $n$ distinct complex numbers, usually called nodes. Vandermonde matrix $V_n$ is defined as the $n \times n$ matrix whose generic element $v_n(i,j)$ is given by

$$v_n(i,j) = x_i^{j-1}, \quad i, j = 1, 2, \ldots, n.$$ (1)

First we introduce some functions and prove their properties relevant in our inversion formula.
Definition 1. Let $\sigma(m,s)$ be the function recursively defined as follows:

\[
\sigma(m,s) = \sigma(m - 1, s) + x_m \sigma(m - 1, s - 1), \quad m, s \text{ integer},
\]

\[
\sigma(m,0) = 1, \quad m = 0, 1, 2, \ldots, \tag{2}
\]

\[
(s < 0) \lor (m < 0) \lor (s > m) \rightarrow \sigma(m,s) = 0.
\]

It is easy to show that the generating function of $\sigma(m,s)$ is:

\[
S_m(x) = \prod_{i=1}^{m} (x - x_i) = \sum_{r=0}^{m} (-1)^{r+m} \sigma(m,m-r)x^r. \tag{3}
\]

Therefore $\sigma(m,s)$ is the $s$th order elementary symmetric function associated to the set $\{x_1,x_2,\ldots,x_m\}$ that is the sum of all products of $s$ distinct nodes chosen from $X_m$. By (3) we have:

\[
\sum_{r=0}^{n} (-1)^r x_i^r \sigma(n,n-r) = 0, \quad i = 1, 2, \ldots, n, \tag{4}
\]

\[
\sum_{r=0}^{n} (-1)^r x_{n+1}^r \sigma(n,n-r) = (-1)^n \prod_{s=1}^{n} (x_{n+1} - x_s). \tag{5}
\]

Definition 2. Let $\phi(m,s)$ be the function recursively defined as follows:

\[
\phi(m+1,s) = \frac{\phi(m,s)}{x_{m+1} - x_s}, \quad m \text{ integer}; \quad s = 1, 2, \ldots, m,
\]

\[
\phi(m+1,m+1) = \prod_{k=1}^{m} \frac{1}{x_{m+1} - x_k}, \tag{6}
\]

\[
\phi(2,1) = \phi(2,2) = \frac{1}{x_2 - x_1}.
\]

It is easy to show that

\[
S'_m(x_k) = (-1)^{m+k} \frac{1}{\phi(m,k)}, \quad k = 1, 2, \ldots, m. \tag{7}
\]

Definition 3. Let $\rho(m,i,j)$ be the function defined as:

\[
\rho(m,i,j) = \sum_{k=1}^{m} \sum_{r=0}^{m-k} (-1)^{k+r} x_i^{k-1} x_j^r \sigma(m,m-k-r), \quad i, j = 1, 2, \ldots, m. \tag{8}
\]
By (2) we have:

$$q(m + 1, i, j) = \sum_{k=1}^{m+1} \sum_{r=0}^{m+1-k} (-1)^{k+r} x_i^{k-1} x_j^r \sigma(m, m + 1 - k - r)$$

$$+ x_{m+1} \sum_{k=1}^{m+1} \sum_{r=0}^{m+1-k} (-1)^{k+r} x_i^{k-1} x_j^r \sigma(m, m - k - r),$$

$$i, j = 1, 2, \ldots, m + 1. \quad (9)$$

Finally, taking into account the properties of $\sigma(m, s)$ in (2) we get the following completely equivalent expressions:

$$q(m + 1, i, j) = (x_{m+1} - x_i) q(m, i, j) - \sum_{r=0}^{m} (-1)^r x_j^r \sigma(m, m - r),$$

$$i, j = 1, 2, \ldots, m + 1,$$ \quad (10)

$$q(m + 1, i, j) = (x_{m+1} - x_j) q(m, i, j) - \sum_{r=0}^{m} (-1)^r x_i^r \sigma(m, m - r),$$

$$i, j = 1, 2, \ldots, m + 1.$$ \quad (11)

The following theorem holds:

**Theorem 1.** Let $V_n$ be a Vandermonde matrix and $W_n$ be its inverse. Then the generic element $w_n(i, j)$ of $W_n$ is:

$$w_n(i, j) = \phi(n, j) \psi(n, i, j), \quad i, j = 1, 2, \ldots, n,$$ \quad (12)

where

$$\psi(n, i, j) = (-1)^{i+j} \sum_{r=0}^{n-i} (-1)^r x_j^r \sigma(n, n - i - r), \quad i, j = 1, 2, \ldots, n.$$ \quad (13)

**Proof.** Let $\theta(n, i, j)$ be the quantity defined as:

$$\theta(n, i, j) = \sum_{k=1}^{n} v_n(i, k) w_n(k, j), \quad i, j = 1, 2, \ldots, n.$$ \quad (14)

The theorem is proved if the following relationship holds:

$$\theta(n, i, j) = \delta_{i,j}, \quad i, j = 1, 2, \ldots, n,$$ \quad (15)

where $\delta_{p,q}$ is the well-known Kronecker $\delta$-function. Taking into account (1), (12) and (8) we obtain:

$$\theta(n, i, j) = (-1)^i \phi(n, j) \rho(n, i, j), \quad i, j = 1, 2, \ldots, n.$$ \quad (16)
Now the following propositions are straightforward:

\[(4) \land (6) \land (11) \rightarrow \theta(n + 1, i, j) = \theta(n, i, j), \tag{17}\]
\[(4) \land (10) \rightarrow \theta(n + 1, n + 1, j) = 0, \tag{18}\]
\[(4) \land (11) \rightarrow \theta(n + 1, i, n + 1) = 0, \tag{19}\]
\[(5) \land (6) \land (10) \rightarrow \theta(n + 1, n + 1, n + 1) = 0, \tag{20}\]

\[i, j = 1, 2, \ldots, n; \quad n = 1, 2, \ldots\]

From (17)–(20) we have:

\[
[\theta(n, i, j) = \delta_{i,j}, \quad i, j = 1, 2, \ldots, n]
\rightarrow [\theta(n + 1, i, j) = \delta_{i,j}, \quad i, j = 1, 2, \ldots, n + 1] \tag{21}\]

and by inspection:

\[
\theta(2, i, j) = (-1)^j \frac{x_i + x_j - x_1 - x_2}{x_2 - x_1} = \delta_{i,j}, \quad i, j = 1, 2. \tag{22}\]

Summing up we get:

\[(21) \land (22) \rightarrow \theta(n, i, j) = \delta_{i,j}, \quad i, j = 1, 2, \ldots, n; \quad n = 2, 3, \ldots \tag{23}\]

Thus (15) is proved by induction and therefore (12) is proved as well.

\[\square\]

3. Computational aspects

As regards the computational aspects of the proposed inversion formula, it is convenient to relate directly the values of the elements of the inverse relevant to the \((i-1)\)th row to those of the \(i\)th row and then to proceed by recursion.

**Proposition 1.** The function \(\psi(n, i, j)\) in (13) can be recursively computed as:

\[
\psi(n, i - 1, j) = x_j\psi(n, i, j) - (-1)^{i+j}\sigma(n, n + 1 - i),
\psi(n, n, j) = (-1)^{n+j}, \quad i = n, n - 1, \ldots, 2; \quad j = 1, 2, \ldots, n. \tag{24}\]

**Proof.** The proof follows by standard algebraic details. \[\square\]

From (24) the following recursion on \(w_n\) holds:

\[
w_n(i - 1, j) = x_jw_n(i, j) - (-1)^{i+j}w_n(n, j)\sigma(n, n + 1 - i),
i = n, n - 1, \ldots, 2; \quad j = 1, 2, \ldots, n, \tag{25}\]
\[
w_n(n, j) = (-1)^{n+j}\phi(n, j), \quad j = 1, 2, \ldots, n.\]
Proposition 2. Eq. (13) is equivalent to:

\[ \psi(n, i, j) = (-1)^{i+j}v_j(n, n-i), \quad i, j = 1, 2, \ldots, n, \]  

(26)

where \( v_j(n, n-i) \) is the \( (n-i) \)th order elementary symmetric function associated to the set \( X_n - \{x_j\} \).

Proof. It must be shown that (26) satisfies the recursion (24), which is equivalent to show that

\[ P_n(i, j) \triangleq x_jv_j(n, n-i) + v_j(n, n+1-i) - \sigma(n, n+1-i) = 0, \]

\[ i, j = 1, 2, \ldots, n. \]

(27)

Now Eq. (27) follows from the easily proved properties of \( P_n(i, j) \):

\[ P_n(n, j) = 0, \quad j = 1, 2, \ldots, n, \]

\[ P_{n+1}(i, j) = x_{n+1}P_n(i, j) + P_n(i-1, j), \quad i = 2, 3, \ldots, n; \quad j = 1, 2, \ldots, n \]

and (26) is proved. □

By using (24) and (26) we propose two algorithms for the inverse of Vandermonde matrix, summarized as follows:

**Algorithm PEF**

1. Compute the quantities \( \sigma(n, s) \), \( s = 0, 1, \ldots, n \) by (2).
2. Compute the quantities \( \phi(n, j) \), \( j = 1, 2, \ldots, n \) by (6).
3. Compute \( \psi(n, i, j) \), \( i, j = 1, 2, \ldots, n \) by recursion (24).
4. Compute the \( j \)th column \( [\psi(n, i, j)\phi(n, j)]_{i \leq i \leq n} \), \( j = 1, 2, \ldots, n \).

**Algorithm EF**

1. Compute the quantities \( \phi(n, j) \), \( j = 1, 2, \ldots, n \) by (6).
2. Compute \( \psi(n, i, j) \), \( i, j = 1, 2, \ldots, n \) by using (26).
3. Compute the \( j \)th column \( [\psi(n, i, j)\phi(n, j)]_{i \leq i \leq n} \), \( j = 1, 2, \ldots, n \).

The first algorithm computes all \( n^2 \) entries of \( V_n \) in \( 5.5n^2 + O(n) \) flops, the latter in \( n^3 + O(n^2) \) flops. It is easy to prove that the algorithm PEF is identical to the Parker inversion algorithm and this is confirmed by all numerical experiments. However a simple modification of PEF algorithm allow us to derive a modified Parker algorithm (PM) that seems to be better than Parker’s one. From Eq. (13) it turns out that for \( i = 1, 2 \) the \( \psi(n, i, j) \) can be directly expressed as:
\[
\psi(n, 1, j) = (-1)^{j+1} \frac{\sigma(n, n)}{x_j^n}, \quad j = 1, 2, \ldots, n, \\
\psi(n, 2, j) = -\psi(n, 1, j) \sum_{q=1, q \neq j}^{n} \frac{1}{x_q}, \quad j = 1, 2, \ldots, n.
\]

Although the PM algorithm differs from the Parker algorithm only in this apparently "nonessential detail" [12], our numerical experiments show that the small difference between the two algorithms is meaningful, from the numerical point of view.

4. Numerical properties

In this section we give representative numerical examples, comparing the accuracy of the following algorithms for the inverse of Vandermonde matrices:

- **BP**
  The Björck–Pereyra algorithm, applied to solving \( n \) linear systems, using the columns of identity matrix for the right-hand sides.

- **Parker**
  The Parker inversion algorithm.

- **PM, EF**
  The proposed inversion algorithms.

The numerical experiments are aimed at comparing the proposed inversion formula with the existing ones in terms of the computational efforts and relative error. All the above algorithms were performed on a Pentium IV for which the unit roundoff is \( u = 2^{-53} \approx 1.11 \times 10^{-16} \) and they have been implemented in Matlab [14]. Their outputs have been migrated into the Mathematica package [15], which allows arbitrary precision numbers, in order to compare them with the “exact” ones computed by using 128 digit precision numbers. We choose, as in [12] the following extended precision relative errors:

\[
e_2 = \frac{\|W_e - W_n\|_2}{\|W_e\|_2}.
\]

Moreover we test our algorithms by the following errors index as in [16]:

\[
e_L = \frac{\|W_n V - I_n\|_\infty}{\|W_n V\|_\infty},
\]

\[
e_R = \frac{\|W_n V - I_n\|_\infty}{\|W_n V\|_\infty},
\]

and introduce another index:

\[
e_\infty = \max_{i,j=1,2,\ldots,n} \left| \frac{W_e(i,j) - W_n(i,j)}{W_e(i,j)} \right|,
\]
where $W_n$ is the inverse of $V$ computed in extended precision and $W_n$ is the inverse computed in low precision by each of the four compared algorithms. In (29) and (30) $I_n$ is the identity matrix of order $n$ and $\|A\|_\infty$ is defined as:

$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |A(i,j)|.$$  

For many authors the numerical behavior of many algorithms, related to polynomial and rational interpolation problems, depends upon the ordering of interpolation points. Therefore we consider the following three orderings of $x_k$:

- **Increasing ordering.** The points are ordered so that $x_1 < x_2 < \cdots < x_n$.
- **Decreasing ordering.** The points are ordered so that $x_1 > x_2 > \cdots > x_n$.
- **Leja ordering.** The points $x_i$ are reordered so that

$$x_1 = \max_{1 \leq k \leq n} |x_k|$$

and

$$\prod_{j=1}^{k-1} |x_k - x_j| = \max_{1 \leq k \leq l \leq n} |x_l - x_j|.$$  

Tables 1–6 report the results for the inverse on random nodes. Two set of experiments have been run for $n = 5, 10, 20$. In the first we have generated

<table>
<thead>
<tr>
<th>$e_2$</th>
<th>BP</th>
<th>EF</th>
<th>Succ. rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand.</td>
<td>Max</td>
<td>2.77–14</td>
<td>3.47–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>2.57–16</td>
<td>1.18–16</td>
</tr>
<tr>
<td>Incr.</td>
<td>Max</td>
<td>4.11–16</td>
<td>3.52–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1.28–16</td>
<td>1.17–16</td>
</tr>
<tr>
<td>Decr.</td>
<td>Max</td>
<td>4.64–16</td>
<td>3.64–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1.44–16</td>
<td>1.18–16</td>
</tr>
<tr>
<td>Leja</td>
<td>Max</td>
<td>2.19–13</td>
<td>4.09–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>2.75–16</td>
<td>1.18–16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P</th>
<th>PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand.</td>
<td>Max</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td>Incr.</td>
<td>Max</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td>Decr.</td>
<td>Max</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td>Leja</td>
<td>Max</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
</tr>
</tbody>
</table>
10000 set of random nodes uniformly distributed in the interval $[0, 1]$. In the second experiments the nodes were distributed in the interval $[-1, 1]$. Tables report the maximum and mean value of (28) and also the fraction of trials in

<table>
<thead>
<tr>
<th>$e_2$</th>
<th>BP</th>
<th>EF</th>
<th>Succ. rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand.</td>
<td>Max 5.49–12</td>
<td>6.64–16</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>Mean 7.11–15</td>
<td>1.82–16</td>
<td></td>
</tr>
<tr>
<td>Incr.</td>
<td>Max 7.05–16</td>
<td>5.92–16</td>
<td>0.53</td>
</tr>
<tr>
<td></td>
<td>Mean 1.90–16</td>
<td>1.81–16</td>
<td></td>
</tr>
<tr>
<td>Decr.</td>
<td>Max 7.00–16</td>
<td>7.27–16</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>Mean 2.05–16</td>
<td>1.83–16</td>
<td></td>
</tr>
<tr>
<td>Leja</td>
<td>Max 4.97–13</td>
<td>6.06–16</td>
<td>0.86</td>
</tr>
<tr>
<td></td>
<td>Mean 1.10–15</td>
<td>1.81–16</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$e_2$</th>
<th>BP</th>
<th>EF</th>
<th>Succ. rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand.</td>
<td>Max 9.77–16</td>
<td>8.34–16</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>Mean 2.14–16</td>
<td>2.12–16</td>
<td></td>
</tr>
<tr>
<td>Incr.</td>
<td>Max 6.82–16</td>
<td>6.69–16</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>Mean 2.13–16</td>
<td>2.11–16</td>
<td></td>
</tr>
<tr>
<td>Decr.</td>
<td>Max 1.13–15</td>
<td>1.02–15</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>Mean 2.26–16</td>
<td>2.23–16</td>
<td></td>
</tr>
<tr>
<td>Leja</td>
<td>Max 6.81–16</td>
<td>6.61–16</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>Mean 2.10–16</td>
<td>2.08–16</td>
<td></td>
</tr>
</tbody>
</table>

Table 2

10 nodes in $[0, 1]$; 10000 runs

<table>
<thead>
<tr>
<th>$e_2$</th>
<th>BP</th>
<th>EF</th>
<th>Succ. rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand.</td>
<td>Max 1.11–08</td>
<td>9.52–16</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>Mean 4.11–12</td>
<td>2.65–16</td>
<td></td>
</tr>
<tr>
<td>Incr.</td>
<td>Max 1.04–15</td>
<td>1.01–15</td>
<td>0.48</td>
</tr>
<tr>
<td></td>
<td>Mean 2.58–16</td>
<td>2.64–16</td>
<td></td>
</tr>
<tr>
<td>Decr.</td>
<td>Max 9.24–16</td>
<td>1.14–15</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>Mean 2.74–16</td>
<td>2.67–16</td>
<td></td>
</tr>
<tr>
<td>Leja</td>
<td>Max 2.59–12</td>
<td>1.02–15</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>Mean 3.00–15</td>
<td>2.65–16</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$e_2$</th>
<th>BP</th>
<th>EF</th>
<th>Succ. rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand.</td>
<td>Max 1.03–15</td>
<td>1.03–15</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>Mean 2.91–16</td>
<td>2.91–16</td>
<td></td>
</tr>
<tr>
<td>Incr.</td>
<td>Max 9.79–16</td>
<td>9.50–16</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>Mean 2.88–16</td>
<td>2.88–16</td>
<td></td>
</tr>
<tr>
<td>Decr.</td>
<td>Max 1.37–15</td>
<td>1.32–15</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>Mean 3.06–16</td>
<td>3.06–16</td>
<td></td>
</tr>
<tr>
<td>Leja</td>
<td>Max 1.00–15</td>
<td>1.00–15</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>Mean 2.86–16</td>
<td>2.86–16</td>
<td></td>
</tr>
</tbody>
</table>

Table 3

20 nodes in $[0, 1]$; 10000 runs
which algorithm EF and PM gives a more accurate result than Björck–Pereyra and Parker–Traub ones, respectively. We not report the results about the other indexes (29)–(31) but they are available.

Table 4
5 nodes in $[-1,1]$; 10000 runs

<table>
<thead>
<tr>
<th>$e_2$</th>
<th>BP</th>
<th>EF</th>
<th>Succ. rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand.</td>
<td>Max</td>
<td>9.44–14</td>
<td>3.45–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>2.64–16</td>
<td>1.05–16</td>
</tr>
<tr>
<td>Incr.</td>
<td>Max</td>
<td>9.45–16</td>
<td>3.82–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1.70–16</td>
<td>1.12–16</td>
</tr>
<tr>
<td>Decr.</td>
<td>Max</td>
<td>8.78–16</td>
<td>3.48–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1.72–16</td>
<td>1.05–16</td>
</tr>
<tr>
<td>Leja</td>
<td>Max</td>
<td>1.06–13</td>
<td>3.84–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>2.59–16</td>
<td>1.02–16</td>
</tr>
</tbody>
</table>

P       | PM
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand.</td>
<td>Max</td>
<td>1.18–15</td>
<td>7.86–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1.32–16</td>
<td>1.22–16</td>
</tr>
<tr>
<td>Incr.</td>
<td>Max</td>
<td>1.20–15</td>
<td>7.95–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1.59–16</td>
<td>1.41–16</td>
</tr>
<tr>
<td>Decr.</td>
<td>Max</td>
<td>1.42–15</td>
<td>8.36–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1.52–16</td>
<td>1.33–16</td>
</tr>
<tr>
<td>Leja</td>
<td>Max</td>
<td>4.84–16</td>
<td>4.85–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1.16–16</td>
<td>1.12–16</td>
</tr>
</tbody>
</table>

Table 5
10 nodes in $[-1,1]$; 10000 runs

<table>
<thead>
<tr>
<th>$e_2$</th>
<th>BP</th>
<th>EF</th>
<th>Succ. rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand.</td>
<td>Max</td>
<td>6.59–12</td>
<td>8.96–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>8.30–15</td>
<td>1.80–16</td>
</tr>
<tr>
<td>Incr.</td>
<td>Max</td>
<td>6.14–15</td>
<td>1.61–15</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>5.59–16</td>
<td>3.45–16</td>
</tr>
<tr>
<td>Decr.</td>
<td>Max</td>
<td>4.62–15</td>
<td>1.76–15</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>5.61–16</td>
<td>3.42–16</td>
</tr>
<tr>
<td>Leja</td>
<td>Max</td>
<td>3.64–13</td>
<td>5.34–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>9.01–16</td>
<td>1.65–16</td>
</tr>
</tbody>
</table>

P       | PM
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand.</td>
<td>Max</td>
<td>2.43–15</td>
<td>2.21–15</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>2.21–16</td>
<td>2.19–16</td>
</tr>
<tr>
<td>Incr.</td>
<td>Max</td>
<td>1.02–14</td>
<td>8.11–15</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>5.77–16</td>
<td>5.64–16</td>
</tr>
<tr>
<td>Decr.</td>
<td>Max</td>
<td>8.17–15</td>
<td>7.23–15</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>5.73–16</td>
<td>5.61–16</td>
</tr>
<tr>
<td>Leja</td>
<td>Max</td>
<td>8.32–16</td>
<td>7.29–16</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>1.80–16</td>
<td>1.80–16</td>
</tr>
</tbody>
</table>
5. Conclusions

In our opinion, the proposed algorithm is competitive in terms of both numerical accuracy and computational effort. The EF algorithm is better than BP for the probability and classic indexes. The $n^3$ flops is the best improvement of $2.5n^3$ flops typical of all algorithms from 1970s [3]. The proposed algorithm comes from an original inversion formula which is obviously equivalent to the existing ones, but is more flexible and allows to obtain special algorithms for particular set of nodes [17–20]. The present results seem to be a good starting point for further analytic works in many problems directly connected to the inverse of the Vandermonde matrix. Our formula is obviously useful when closed expressions of $\sigma(n,q)$ and $\phi(n,j)$ are known. Here we report such formulas for different set of nodes that are of practical interest in the applications [21]:

- Equidistant nodes in $[1,n]$: $\{x_k = k, k = 1, 2, \ldots, n\}$

\[
\phi(n,j) = \frac{1}{(n-j)!(j-1)!}, \quad j = 1, 2, \ldots, n, \quad (32)
\]

\[
\sigma(n,q) = \begin{bmatrix} n+1 \\ n+1-q \end{bmatrix}, \quad q = 0, 1, \ldots, n, \quad (33)
\]

where $\binom{i}{j}$ are the Stirling numbers of the first kind [9].
• Equidistant nodes in $[0, 1]$: \( \{x_k = \frac{k-1}{n-1}, \ k = 1, 2, \ldots, n\} \)

\[
\phi(n, j) = \frac{(n-1)^{n-1}}{(n-j)!(j-1)!}, \quad j = 1, 2, \ldots, n, \\
\sigma(n, q) = \frac{1}{(n-1)^2} \left[ \begin{array}{c} n \\ n-q \end{array} \right], \quad q = 0, 1, \ldots, n.
\] (34, 35)

• Equidistant nodes in $[-1, 1]$: \( \{x_k = \frac{2k-n-1}{n-1}, \ k = 1, 2, \ldots, n\} \)

\[
\phi(n, j) = \frac{(\frac{n-1}{2})^{n-1}}{(n-j)!(j-1)!}, \quad j = 1, 2, \ldots, n, \\
\sigma(n, q) = \left( \frac{2}{n-1} \right)^q \sum_{k=0}^{q} (-1)^k \left( \frac{n+1}{2} \right)^k \left( \frac{n+k-q}{k} \right) \left[ \begin{array}{c} n+1 \\ n+1+k-q \end{array} \right], \\
q = 0, 1, \ldots, n. 
\] (36, 37)

• Chebyshev nodes: \( \{x_k = \cos\left[\frac{k-1}{2n}\pi\right], \ k = 1, 2, \ldots, n\} \)

\[
\phi(n, j) = (-1)^{n+1} \frac{2^{n-1}}{n} \sin\left(\frac{2j-1}{2n}\pi\right), \quad j = 1, 2, \ldots, n, \\
\sigma(n, 2q) = \frac{(-1)^q}{2^{2q}} \frac{1}{\cos\left(\frac{\pi}{2n}\right)^{2q}} \left( \frac{n}{2q} \right) \left( \frac{q}{n-1} \right), \quad q = 0, 1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor. 
\] (38, 39)

• Extended Chebyshev nodes: \( \{x_k = -\frac{\cos\left(\frac{k-1}{n}\pi\right)}{\cos\left(\frac{\pi}{2n}\right)}, \ k = 1, 2, \ldots, n\} \)

\[
\phi(n, j) = \frac{2^{n-1}}{n} \cos\left(\frac{\pi}{2n}\right)^{n-1} \sin\left(\frac{2j-1}{2n}\pi\right), \quad j = 1, 2, \ldots, n, \\
\sigma(n, 2q) = \frac{(-1)^q}{2^{2q}} \frac{1}{\cos\left(\frac{\pi}{2n}\right)^{2q}} \left( \frac{n}{2q} \right) \left( \frac{q}{n-1} \right), \quad q = 0, 1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor. 
\] (40, 41)

• Gauss–Lobatto Chebyshev (extrema): \( \{x_k = -\cos\left[\frac{k-1}{n}\pi\right], \ k = 1, 2, \ldots, n\} \)

\[
\phi(n, j) = \begin{cases} 
\frac{2^{n-3}}{n-1}, & j = 1, n, \\
\frac{2^{n-2}}{n-1}, & j = 2, \ldots, n-1,
\end{cases} \\
\sigma(n, 2q) = (-1)^q \frac{1}{2^{2q}} \left( \frac{n-q}{q} \right) \frac{n^2-n-2q}{(n-q-1)(n-q)}, \quad q = 0, 1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor. 
\] (42, 43)
Fekete nodes: such nodes are roots of the polynomial
\[ P_n(x) = (x^2 - 1)L_{n-1}'(x), \tag{44} \]
where \( L_n(x) \) is the \( n \)th Legendre polynomial in \( x \).

\[ \sigma(n, 2q) = (-1)^q \frac{(n-2)(q-1)}{(2n-3)(2q-1)} \binom{n}{2q}, \quad q = 0, 1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor. \tag{45} \]

Unfortunately the function \( \phi(n,j) \) is not rational for \( j = 2, 3, \ldots, n-1 \).

Acknowledgement

We wish to thank Prof. L. Carotenuto of Universitá della Calabria, Cosenza, Italy, for the helpful discussions.

References