

A Note on a Recent Paper by J. S. Respondek

Moawwad E. A. El-Mikkawy

Mathematics Department, Faculty of Science, Mansoura University, Mansoura, Egypt
 Email: m_elmikkawy@yahoo.com

Received February 9, 2012; revised April 26, 2012; accepted May 4, 2012

ABSTRACT

In this note, we give comments on a very recent paper by J. S. Respondek [1]. In [1], the author claims that an algorithm in [2] contains a severe error. We show that the algorithm in [2] can be implemented properly without causing any errors by using vectors (one-dimensional arrays) rather than using 2-dimensional arrays. To enable users and programmers of the algorithm to carry out the computations using all existing subscripts and superscripts in the algorithm, we give a correction in the first line of the algorithm. A Maple implementation for the algorithm, as it is in [2], is given as an example for symbolic programming.

Keywords: Elementary Symmetric Function; Algorithm; Maple; Symbolic Programming

1. Introduction and Objectives

For convenience of the reader, we begin this section by introducing the following abstract from [1] concerning the algorithm in [2].

“In this paper, we give the comments on the article ‘Inversion of a Generalized Vandermonde Matrix’ by M. E. A. El Mikkawy, Int. J. Computer Math. 80 (2003), pp. 759-765. The article gives an algorithm for the elementary symmetric function’s calculation which contains a severe error. In these comments, we have proposed necessary corrections of that algorithm”.

Let us begin our discussion by giving the following definition and the algorithm for the elementary symmetric function’s in [2]:

Definition 1.1 ([2], p. 760). If the n parameters c_1, c_2, \dots, c_n are distinct, then the elementary symmetric functions $\sigma_{i,j}^{(n)}$ in $c_1, c_2, \dots, c_{j-1}, c_{j+1}, \dots, c_n$ are defined for $j = 1(1)n$ by:

$$\sigma_{1,j}^{(n)} = 1$$

$$\sigma_{i,j}^{(n)} = \sum_{\substack{\eta_1=1 \\ \eta_1 \neq j}}^n \sum_{\substack{\eta_2=1 \\ \eta_2 \neq j}}^n \cdots \sum_{\substack{\eta_{i-1}=i-2+1 \\ \eta_{i-1} \neq j}}^n \prod_{m=1}^{i-1} c_{\eta_m} \quad \text{for } i = 2(1)n. \quad (1.1)$$

Denote by:

$$\sigma_n^{(G)} = \left(\sigma_{i,j}^{(n)} \right)_{1 \leq i,j \leq n} \quad (1.2)$$

Algorithm (Algorithm 2.1 in [2], p. 760). For $n \geq 1$, we may calculate the elements of the first column of the $n \times n$ matrix $\sigma_n^{(G)}$ in (1.2) as follows:

Set $\sigma_{1,1}^{(1)} = 1$
 For $i = 2, 3, \dots, n$

$$\sigma_{i,1}^{(i)} = \sigma_{i-1,1}^{(i-1)} c_i \quad (1.3)$$

For $j = i-1, i-2, \dots, 2$

$$\sigma_{j,1}^{(i)} = \sigma_{j-1,1}^{(i-1)} c_i + \sigma_{j,1}^{(i-1)}$$

Next j

Next i

The elements in the remaining $n-1$ columns of $\sigma_n^{(G)}$ may be obtained by symmetry using

$$\sigma_{i,k}^{(n)} = \sigma_{i,1}^{(n)} \Big|_{c_k \rightarrow c_1}, \quad i = 1(1)n, \quad k = 2(1)n. \quad (1.4)$$

The notation in (1.4) means that for specific i and k , $\sigma_{i,k}^{(n)}$ may be obtained from the algebraic expression of $\sigma_{i,1}^{(n)}$ by replacing each c_k by c_1 in the expression of $\sigma_{i,1}^{(n)}$.

In [1], Respondek claims that the above algorithm contains a severe error. In the next section, we are going to show that the algorithm, as it is in [2] and (1.3), can be implemented properly without causing any errors by using vectors (one-dimensional arrays) rather than using 2-dimensional arrays. To enable users and programmers of the algorithm to carry out the computations using all existing subscripts and superscripts in the algorithm, we give a correction in the first line of the algorithm. A Maple implementation for the algorithm, as it is in [2] and (1.3), is given as an example for symbolic programming.

2. Main Results

It is now time to show that the algorithm, as it is in [2] and (1.3), can be implemented properly without causing

any errors by using vectors (one-dimensional arrays) rather than using 2-dimensional arrays. The following is a Maple implementation for the algorithm in [2] and (1.3). Some sample output for the case $n = 4$ is given.

```
> # Implementation of Algorithm 2.1 in [2], p. 760
> restart:
> n:=4: g :=array(1..n):
> g[1]:=1:
> for i from 2 to n do
  g[i]:= c[i] * g[i-1]:
  for j from i-1 to 2 by -1 do
```

```
    g[j] := simplify(g[j] + c[i] * g[j-1]);
  od:
od:
> sigma := array(1..n,1..n):
> for i to n do sigma[i,1] := g[i] ; od:
> for i to n do
  for j from 2 to n do
    sigma[i,j] := subs({c[j]=c[1]},sigma[i,1]);
  od:
od:
> sigma = op(sigma);
```

$$\sigma = \begin{bmatrix} 1 & 1 & 1 & 1 \\ c_2 + c_3 + c_4 & c_1 + c_3 + c_4 & c_2 + c_1 + c_4 & c_2 + c_3 + c_1 \\ c_3c_2 + c_4c_2 + c_4c_3 & c_3c_1 + c_4c_1 + c_4c_3 & c_1c_2 + c_4c_2 + c_4c_1 & c_3c_2 + c_1c_2 + c_3c_1 \\ c_4c_3c_2 & c_4c_3c_1 & c_4c_1c_2 & c_1c_3c_2 \end{bmatrix}$$

To enable users and programmers of the algorithm to carry out the computations using all existing subscripts and superscripts in the algorithm, as actually done by J. S. Respondek [1], we give a correction in the first line of the algorithm. The modified algorithm is now given by:

A Modified Algorithm. For $n \geq 1$ we may calculate the elements of the first column of the $n \times n$ matrix $\sigma_n^{(G)}$ in (1.2) as follows:

Set $\sigma_{1,1}^{(i)} = 1, i = 1(1) n - 1.$

For $i = 2, 3, \dots, n$

$$\sigma_{i,1}^{(i)} = \sigma_{i-1,1}^{(i-1)} c_i \tag{2.1}$$

For $j = i - 1, i - 2, \dots, 2$

$$\sigma_{j,1}^{(i)} = \sigma_{j-1,1}^{(i-1)} c_i + \sigma_{j,1}^{(i-1)}$$

```
    Next j
  Next i
```

REFERENCES

- [1] J. S. Respondek, "Comments on 'Inversion of a Generalized Vandermonde Matrix' by M. E. A. El-Mikkawy," *International Journal of Computer Mathematics*, Vol. 88, No. 1, 2011, pp. 3565-3568. [doi:10.1080/00207160.2011.603413](https://doi.org/10.1080/00207160.2011.603413)
- [2] M. E. A. El-Mikkawy, "Inversion of a Generalized Vandermonde Matrix by M. E. A. El-Mikkawy," *International Journal of Computer Mathematics*, Vol. 80, 2003, pp. 759-765. [doi:10.1080/0020716021000059133](https://doi.org/10.1080/0020716021000059133)