SOME DETERMINANTAL IDENTITIES IN A VECTOR SPACE, WITH APPLICATIONS

Claude Brezinski
Laboratoire d'Analyse Numérique et d'Optimisation
Université des Sciences et Techniques de Lille / France

***

Work performed under the NATO Research Grant 027.81

***

ABSTRACT :

The determinantal identities of Al. Magnus, J.J. Sylvester and F.F. Schweins are extended to determinants whose first row consists of elements of a vector space and whose other rows are formed by scalars. These identities are then used to derive a recursive algorithm having many applications.

***

People working on Padé approximants or on nonlinear sequence transformations are familiar with the determinantal identities named after Al. Magnus, J.J. Sylvester and F.F. Schweins.

In the first part of this paper I shall extend these identities to determinants whose first row consists of elements of a vector space E over a field K and whose other rows are formed by elements of K. The same proofs can, of course, be used in the classical case thus providing a demonstration that I believe to be new for Sylvester's and Schweins' identities.

In the second part of this paper a recursive algorithm will be derived from the extended Sylvester's identity. This algorithm, called the recursive projection algorithm (RPA), has many applications that will be studied in a forthcoming paper.

1 - Magnus' identity

Let E be a vector space over a field K. Let $x_i \in E$ and $a_{ij} \in K$. We consider the generalized determinant

$$\begin{vmatrix} x_o & x_1 & \cdots\cdots & x_k \\ a_{oo} & a_{o1} & \cdots\cdots & a_{ok} \\ \cdots & \cdots & \cdots & \cdots \\ a_{k-1,o} & a_{k-1,1} & \cdots & a_{k-1,k} \end{vmatrix} .$$

This determinant is equal to the element of E obtained by expanding the determinant with respect to its first row by using the classical rule for expanding a determinant.

Let us now consider the ratio

$$\frac{\begin{vmatrix} x_o & x_1 & \cdots\cdots & x_k \\ a_{oo} & a_{o1} & \cdots\cdots & a_{ok} \\ \cdots & \cdots & \cdots & \cdots \\ a_{k-1,o} & a_{k-1,1} & \cdots & a_{k-1,k} \end{vmatrix}}{|A|} = a_o x_o + \ldots + a_k x_k \in E$$

where the matrix A is

$$A = \begin{pmatrix} a_{o1} & \cdots\cdots & a_{ok} \\ \cdots & \cdots & \cdots \\ a_{k-1,1} & \cdots\cdots & a_{k-1,k} \end{pmatrix} .$$

This ratio of determinants is equal to the linear combination $a_o\ x_o + \ldots$ $a_k x_k \in E$, where the a_i's are the solution of the system

$$\begin{pmatrix} 1 & 0 & \cdots\cdots & 0 \\ a_{oo} & a_{o1} & \cdots\cdots & a_{ok} \\ \cdots & \cdots & \cdots & \cdots \\ a_{k-1,o} & a_{k-1,1} & \cdots & a_{k-1,k} \end{pmatrix} \begin{pmatrix} a_o \\ a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

or equivalently to

$$a_o = 1$$

$$\begin{pmatrix} a_{o1} & \cdots\cdots\cdots & a_{ok} \\ \cdots & \cdots\cdots\cdots & \cdots \\ a_{k-1,1} & \cdots\cdots\cdots & a_{k-1,k} \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_k \end{pmatrix} = - \begin{pmatrix} a_{oo} \\ \vdots \\ a_{k-1,o} \end{pmatrix}.$$

We set $a = (a_1, \ldots, a_k)^T$, $u = (a_{oo}, \ldots, a_{k-1,o})^T$ and $x = (x_1, \ldots, x_k)^T$. Thus

$$a = - A^{-1} u.$$

We shall make use of the notation

$$a * x = x * a = a_1 x_1 + \ldots + a_k x_k.$$

Thus

$$a_o x_o + \ldots + a_k x_k = x_o - x * A^{-1} u$$

and we finally obtain the identity

$$\frac{\begin{vmatrix} x_o & x^T \\ u & A \end{vmatrix}}{|A|} = x_o - x * A^{-1} u.$$

This identity is the extension to a vector space of an identity due to Al. Magnus [7, p. i 17].

2 - SYLVESTER'S IDENDITY

We shall now make use of Magnus' identity to derive a generalisation of Sylvester's identity.

We set

$$B = \begin{pmatrix} a_{o1} & \cdots\cdots\cdots & a_{o,k-1} \\ \cdots & \cdots\cdots\cdots & \cdots \\ a_{k-2,1} & \cdots\cdots\cdots & a_{k-2,k-1} \end{pmatrix},$$

$$u'' = (a_{ok}, \ldots, a_{k-2,k})^T \quad , \quad v = (a_{k-1,1}, \ldots, a_{k-1,k-1})^T$$

$$u' = (a_{oo}, \ldots, a_{k-2,o})^T \quad , \quad x' = (x_1, \ldots, x_{k-1})^T.$$

With these notations we have

$$u = \begin{pmatrix} u' \\ a_{k-1,o} \end{pmatrix} , \quad x = \begin{pmatrix} x' \\ x_k \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} B & u'' \\ v^T & a_{k-1,k} \end{pmatrix} .$$

Using the bordering method [4, pp. 110-111] for solving the system $Aa = -u$ we get

$$A^{-1}u = \begin{pmatrix} B^{-1}u' \\ 0 \end{pmatrix} + R \begin{pmatrix} -B^{-1}u'' \\ 1 \end{pmatrix}$$

with

$$R = \frac{a_{k-1,o} - (v,\ B^{-1}u')}{a_{k-1,k} - (v,\ B^{-1}u'')} .$$

Thus

$$x * A^{-1}u = x' * B^{-1}u' + R(-x' * B^{-1}u'' + x_k).$$

Let us now calculate R. Using the scalar Magnus' identity we have

$$a_{k-1,o} - (v,\ B^{-1}u') = \begin{vmatrix} a_{k-1,o} & a_{k-1,1} & \cdots & a_{k-1,k-1} \\ a_{oo} & a_{o1} & \cdots & a_{o,k-1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{k-2,o} & a_{k-2,1} & \cdots & a_{k-2,k-1} \end{vmatrix} / |B|$$

$$= (-1)^{k-1} |A''| / |B|$$

with

$$A'' = \begin{pmatrix} a_{oo} & \cdots & a_{o,k-1} \\ \cdots & \cdots & \cdots \\ a_{k-1,o} & \cdots & a_{k-1,k-1} \end{pmatrix} = \begin{pmatrix} u' & B \\ a_{k-1,o} & v^T \end{pmatrix} .$$

We also have

$$a_{k-1,k} - (v,\ B^{-1}u'') = \begin{vmatrix} a_{k-1,k} & a_{k-1,1} & \cdots & a_{k-1,k-1} \\ a_{ok} & a_{o1} & \cdots & a_{o,k-1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{k-2,k} & a_{k-2,1} & \cdots & a_{k-2,k-1} \end{vmatrix} / |B|$$

$$= |A| / |B|.$$

Thus we get

$$x * A^{-1}u = x' * B^{-1}u' + (-1)^{k-1} \frac{|A''|}{|A|} (x_k - x' * B^{-1} u'')$$

or

$$x_o - x * A^{-1} u = x_o - x' * B^{-1}u' - (-1)^{k-1} \frac{|A''|}{|A|} (x_k - x' * B^{-1}u'').$$

By Magnus' generalized identity we have

$$\begin{vmatrix} x_o & x^T \\ u & A \end{vmatrix} = |A| \; (x_o - x * A^{-1} u)$$

$$\begin{vmatrix} x_o & x'^T \\ u' & B \end{vmatrix} = |B| \; (x_o - x' * B^{-1}u')$$

$$\begin{vmatrix} x_k & x'^T \\ u'' & B \end{vmatrix} = |B| \; (x_k - x' * B^{-1}u'') = (-1)^{k-1} \begin{vmatrix} x'^T & x_k \\ B & u'' \end{vmatrix} .$$

Thus we get

$$\begin{vmatrix} x_o & x^T \\ u & A \end{vmatrix} \; |B| = |A| \begin{vmatrix} x_o & x'^T \\ u' & B \end{vmatrix} - |A''| \begin{vmatrix} x'^T & x_k \\ B & u'' \end{vmatrix}$$

or

$$\begin{vmatrix} x_o & x_1 & \cdots & x_k \\ a_{oo} & a_{o1} & \cdots & a_{ok} \\ \cdots & \cdots & \cdots & \cdots \\ a_{k-1,o} & a_{k-1,1} & \cdots & a_{k-1,k} \end{vmatrix} \begin{vmatrix} a_{o1} & \cdots & a_{o,k-1} \\ \cdots & \cdots & \cdots \\ a_{k-2,1} & \cdots & a_{k-2,k-1} \end{vmatrix} =$$

$$\begin{vmatrix} a_{o1} & \cdots & a_{ok} \\ \cdots & \cdots & \cdots \\ a_{k-1,1} & \cdots & a_{k-1,k} \end{vmatrix} \begin{vmatrix} x_o & x_1 & \cdots & x_{k-1} \\ a_{oo} & a_{o1} & \cdots & a_{o,k-1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{k-2,o} & a_{k-2,1} & \cdots & a_{k-2,k-1} \end{vmatrix}$$

$$- \begin{vmatrix} a_{oo} & \cdots & a_{o,k-1} \\ \cdots & \cdots & \cdots \\ a_{k-1,o} & \cdots & a_{k-1,k-1} \end{vmatrix} \begin{vmatrix} x_1 & \cdots & x_k \\ a_{o1} & \cdots & a_{ok} \\ \cdots & \cdots & \cdots \\ a_{k-2,1} & \cdots & a_{k-2,k} \end{vmatrix} .$$

This is Sylvester's identity. Thus Sylvester's identity consists in solving

$$\begin{pmatrix} B & u'' \\ v & a_{k-1,k} \end{pmatrix} a = - \begin{pmatrix} u' \\ a_{k-1,o} \end{pmatrix}$$

by the bordering method and then calculating $x * A^{-1}u$.

3 - Schweins' identity

Let $c_i \in K$, $c = (c_1, \ldots, c_k)^T$ and $c' = (c_1, \ldots, c_{k-1})^T$. By Sylvester's identity we have

$$\begin{vmatrix} x_o & x'^T & x_k \\ u' & B & u'' \\ c_o & c'^T & c_k \end{vmatrix} |B| = \begin{vmatrix} x_o & x'^T \\ u' & B \end{vmatrix} \begin{vmatrix} B & u'' \\ c'^T & c_k \end{vmatrix} - \begin{vmatrix} x'^T & x_k \\ B & u'' \end{vmatrix} \begin{vmatrix} u' & B \\ c_o & c'^T \end{vmatrix}$$

or

$$\begin{vmatrix} x_o & x'^T & x_k \\ c_o & c'^T & c_k \\ u' & B & u'' \end{vmatrix} |B| = \begin{vmatrix} x_o & x'^T \\ u' & B \end{vmatrix} \begin{vmatrix} c'^T & c_k \\ B & u'' \end{vmatrix} - \begin{vmatrix} x'^T & x_k \\ B & u'' \end{vmatrix} \begin{vmatrix} c_o & c'^T \\ u' & B \end{vmatrix} \qquad (1).$$

Moreover

$$\begin{vmatrix} x_o & x^T \\ u & A \end{vmatrix} |B| = |A| \begin{vmatrix} x_o & x'^T \\ u' & B \end{vmatrix} - |A''| \begin{vmatrix} x'^T & x_k \\ B & u'' \end{vmatrix} \qquad (2)$$

$$\begin{vmatrix} c_o & c^T \\ u & A \end{vmatrix} |B| = |A| \begin{vmatrix} c_o & c'^T \\ u' & B \end{vmatrix} - |A''| \begin{vmatrix} c'^T & c_k \\ B & u'' \end{vmatrix} \qquad (3).$$

Multiplying (2) by $\begin{vmatrix} c_o & c'^T \\ u' & B \end{vmatrix}$, (1) by $|A''|$ and subtracting we get

$$|B| \begin{vmatrix} x_o & x^T \\ u & A \end{vmatrix} \begin{vmatrix} c_o & c'^T \\ u' & B \end{vmatrix} - |B|\ |A''| \begin{vmatrix} x_o & x'^T & x_k \\ c_o & c'^T & c_k \\ u' & B & u'' \end{vmatrix}$$

$$= \begin{vmatrix} x_o & x'^T \\ u' & B \end{vmatrix} \left\{ |A| \begin{vmatrix} c_o & c'^T \\ u' & B \end{vmatrix} - |A''| \begin{vmatrix} c'^T & c_k \\ B & u'' \end{vmatrix} \right\}$$

$$= \begin{vmatrix} x_o & x'^T \\ u' & B \end{vmatrix} \begin{vmatrix} c_o & c^T \\ u & A \end{vmatrix} |B|$$

by using (3).

Thus if $|B| \neq 0$ we obtain the generalized Schweins' identity

$$\begin{vmatrix} x_o & x^T \\ u & A \end{vmatrix} \begin{vmatrix} c_o & c'^T \\ u' & B \end{vmatrix} - \begin{vmatrix} x_o & x'^T \\ u' & B \end{vmatrix} \begin{vmatrix} c_o & c^T \\ u & A \end{vmatrix} = |A''| \begin{vmatrix} x_o & x'^T & x_k \\ c_o & c'^T & c_k \\ u' & B & u'' \end{vmatrix}.$$

Up to now $|A|$ and $|B|$ have been assumed to be different from zero. This is not a restriction for our purpose since these determinants will appear as denominators. However the three preceding determinantal identities still hold if $|A|$ or/and $|B|$ are equal to zero.

Let $\tilde{A}$ be the adjugate matrix of A that is the matrix formed by the cofactors of A. Then it is easy to see that Magnus'identity can also be written as

$$\begin{vmatrix} x_o & x^T \\ u & A \end{vmatrix} = |A|\ x_o - x * \tilde{A}\, u.$$

If A is regular then $\tilde{A} = |A|\ A^{-1}$ and we get the identity given in the first section.

4 - THE RECURSIVE PROJECTION ALGORITHM

Let E be a vector space and E^* its dual. We shall denote by $<.,.>$ the bilinear form of the duality between E and E^*. Let $y \in E$, $x_i \in E$ and $z_i \in E^*$.

We set

$$N_k = \begin{vmatrix} y & x_1 & \cdots\cdots & x_k \\ <z_1, y> & <z_1, x_1> & \cdots & <z_1, x_k> \\ \cdots & \cdots & \cdots & \cdots \\ <z_k, y> & <z_k, x_1> & \cdots & <z_k, x_k> \end{vmatrix}$$

$$D_k = \begin{vmatrix} <z_1, x_1> & \cdots\cdots & <z_1, x_k> \\ \cdots & \cdots & \cdots \\ <z_k, x_1> & \cdots\cdots & <z_k, x_k> \end{vmatrix}$$

$$N_{k,i} = \begin{vmatrix} x_i & x_1 & \cdots\cdots & x_k \\ <z_1, x_i> & <z_1, x_1> & \cdots & <z_k, x_k> \\ \cdots & \cdots & \cdots & \cdots \\ <z_k, x_i> & <z_k, x_1> & \cdots & <z_k, x_k> \end{vmatrix}$$

$$E_k = N_k / D_k \quad , \quad g_{k,i} = N_{k,i} / D_k$$

We shall now give a recursive algorithm, named the recursive projection algorithm, for computing the E_k's.

We set, in Magnus' extended identity

$$x_o = y$$
$$a_{io} = <z_{i+1}, y> \qquad i = 0, \ldots, k-1$$
$$a_{ij} = <z_{i+1}, x_j> \qquad i = 0, \ldots, k-1 \text{ and } j = 1, \ldots, k.$$

Thus

$$E_k = x_o - x * A^{-1} u$$
$$E_{k-1} = x_o - x' * B^{-1} u'$$
$$g_{k-1,k} = x_k - x' * B^{-1} u''$$

and

$$E_k = E_{k-1} - (-1)^{k-1} \frac{|A''|}{|A\ |} g_{k-1,k}.$$

But

$$|B| <z_k, g_{k-1,k}> = \begin{vmatrix} <z_k, x_k> & <z_k, x'> \\ u'' & B \end{vmatrix} = |A|$$

$$|B| <z_k, E_{k-1}> = \begin{vmatrix} <z_k, y> & <z_k, x'> \\ u' & B \end{vmatrix} = (-1)^{k-1} |A''|$$

and we get

$$E_k = E_{k-1} - (-1)^{k-1} \frac{(-1)^{k-1} |B| <z_k, E_{k-1}>}{|B| <z_k, g_{k-1,k}>} g_{k-1,k}$$

or

$$E_k = E_{k-1} - \frac{<z_k, E_{k-1}>}{<z_k, g_{k-1,k}>} g_{k-1,k} .$$

Since the expression for $g_{k,i}$ is obtained from that of E_k by replacing y by x_i then a similar recursive relation holds for $g_{k,i}$.

Thus we finally obtain the following recursive algorithm

$$E_o = y$$
$$g_{o,i} = x_i \qquad i \geq 1$$

$$E_k = E_{k-1} - \frac{<z_k, E_{k-1}>}{<z_k, g_{k-1,k}>} \; g_{k-1,k} \qquad k > 0 \qquad \text{principal rule}$$

$$g_{k,i} = g_{k-1,i} - \frac{<z_k, g_{k-1,i}>}{<z_k, g_{k-1,k}>} \; g_{k-1,k} \qquad i > k > 0 \quad \text{auxiliary rules}$$

This algorithm, called the recursive projection algorithm (RPA) can be compared with E and MNA algorithms for recursive extrapolation and interpolation which have been studied by various authors [1, 2, 5, 8, 9, 11, 12]. The RPA can also be obtained by applying Sylvester's identity to N_k.

From the computational point of view, the RPA is not a very simple algorithm since it involves a principal rule and auxiliary rules. We shall see now how to obtain a more compact algorithm with only one single rule.

First let us write x_o instead of y to simplify the notations. We set

$$N_k^{(i)} = \begin{vmatrix} x_i & x_{i+1} & \cdots & x_{i+k} \\ <z_1, x_i> & <z_1, x_{i+1}> & \cdots & <z_1, x_{i+k}> \\ \cdots & \cdots & \cdots & \cdots \\ <z_k, x_i> & <z_k, x_{i+1}> & \cdots & <z_k, x_{i+k}> \end{vmatrix}$$

$$D_k^{(i)} = \begin{vmatrix} <z_1, x_{i+1}> & \cdots & <z_1, x_{i+k}> \\ \cdots & \cdots & \cdots \\ <z_k, x_{i+1}> & \cdots & <z_k, x_{i+k}> \end{vmatrix}$$

and

$$e_k^{(i)} = N_k^{(i)} \,/\, D_k^{(i)} .$$

Obviously, since $x_o = y$

$$D_k^{(o)} = D_k, \quad N_k^{(o)} = N_k \quad \text{and} \quad N_k^{(1)} = (-1)^k \, N_{k,k+1}$$

and thus

$$e_k^{(o)} = E_k \quad \text{and} \quad e_k^{(1)} = (-1)^k \, g_{k,k+1} \, D_k^{(o)} \,/\, D_k^{(1)} .$$

We shall now construct a recursive algorithm for computing the e_k's.

Let us apply Sylvester's identity to $N_k^{(i)}$. We get

$$N_k^{(i)} \, D_{k-1}^{(i)} = N_{k-1}^{(i)} \, D_k^{(i)} - N_{k-1}^{(i+1)} \, D_k^{(i-1)} .$$

Dividing both sides by $D_{k-1}^{(i)} \, D_k^{(i)}$ and using the relation

$$<z_k, N_{k-1}^{(i)}> = (-1)^{k-1} \, D_k^{(i-1)}$$

we immediately obtain the algorithm

$$e_o^{(i)} = x_i \qquad i \geq 0$$

$$e_k^{(i)} = e_{k-1}^{(i)} - \frac{<z_k, e_{k-1}^{(i)}>}{<z_k, e_{k-1}^{(i+1)}>} e_{k-1}^{(i+1)} \qquad i, k \geq 0$$

This algorithm will be called the compact recursive projection algorithm (in short the CRPA). It essentially computes the sames quantities as the RPA and thus we shall differentiate both algorithms only when necessary.

The RPA has many applications. It is connected with the conjugate gradient method, with Fourier's expansion, with Rosen's method for recursive projection [10], with Henrici's method for solving systems of nonlinear equations [6], with the E and MNA algorithms, with the general extrapolation and interpolation problems, with the vector E-algorithm, with formal orthogonal polynomials, with least squares extrapolation and with continuous prediction algorithms. These applications and connections will be studied in details in a forthcoming paper [3].

References

[1] C. BREZINSKI
A general extrapolation algorithm
Numer. Math., 35 (1980) 175-187.

[2] C. BREZINSKI
The Mühlbach-Neville-Aitken algorithm and some extensions.
BIT, 20 (1980) 444-451.

[3] C. BREZINSKI
Recursive interpolation, extrapolation and projection.
J. Comp. Appl. Math., submitted.

[4] V.N. FADDEEVA
Computational methods of linear algebra.
Dover, New-York, 1959.

[5] T. HÅVIE
Generalized Neville type extrapolation schemes.
BIT, 19 (1979) 204-213.

[6] P. HENRICI
Elements of numerical analysis.
Wiley, New-York, 1964.

[7] Al. MAGNUS
Fractions continues généralisées : théorie et applications.
Thèse, Université Catholique de Louvain, 1976.

[8] G. MEINARDUS, G.D. TAYLOR
Lower estimates for the error of best uniform approximation.
J. Approx. Theory, 16 (1976) 150-161.

[9] G. MÜHLBACH
The general Neville-Aitken algorithm and some applications.
Numer. Math., 31 (1978) 97-110.

[10] J.B. ROSEN
The gradient projection method for nonlinear programming. Part I.
Linear constraints.
J. SIAM, 8 (1960) 181-217.

[11] C. SCHNEIDER
Vereinfachte Rekursionen zur Richardson-Extrapolation in Spezialfällen.
Numer. Math., 24 (1975) 177-184.

[12] J. WIMP
Sequence transformations and their applications.
Academic Press, New-York, 1981.

UER IEEA - M3
Université de Lille I
59655 - Villeneuve d'Ascq cedex
France