Polynomials arising in factoring generalized Vandermonde determinants III : computation of their roots

Stefano De Marchi
Department of Computer Science, University of Verona, ITALY
E-mail: stefano.demarchi@univr.it

Maria Morandi Cecchi
Department of Pure and Applied Mathematics, University of Padova, ITALY
E-mail: mcecchi@math.unipd.it

Abstract

Determinants of the form

$$V_\alpha(\mathbf{x}) = |x_i^{\alpha_j}|, \quad i,j = 1,\dots,N$$

where $\mathbf{x} = (x_1,\dots,x_N)$ is formed by N distinct points belonging to some interval $[a,b]$ of the real line and the α_j are ordered integers $\alpha_1 > \alpha_2 > \cdots > \alpha_N > 0$ are known as *generalized Vandermonde determinants*. These determinants were considered by Heineman at the end of the 1920s [9]. The paper presents some results concerning univariate polynomials arising from $V_\alpha(\mathbf{x})$, by considering one of the x_i as an unknown. In particular we shall consider the problem of computing their roots by means of a *family of iteration functions* having a symmetric structure which is connected to the structure of our polynomials.

Keywords: Polynomials, determinants, factorization of matrices.

AMS Subject Classification: 11C08,15A23.

1 Introduction

Generalized Vandermonde determinants, shortly GVD, arise in many contexts of applied mathematics. For instance in the framework of linear difference equations with constant coefficients (cf. e.g. [1]) or in two step Runge-Kutta-Nyström collocation method for the numerical integration of second order differential equations (cf. e.g. [13]) or again in the problem of finding generalized Fekete points for an arc of a curve (cf. [3]). These examples simply show the variety of applications in which GVD arise and the necessity of finding a method for computing them. The computation of GVD is almost a known problem and, as we shall see in Section 2.1, every GVD has a factorization in terms of a *classical Vandermonde determinant*, that is

$$VDM(x_1,\dots,x_N) := \prod_{1\le i<j\le N} (x_j - x_i) \tag{1}$$

where the x_i are points belonging to some interval of the real line, and a *Schur functions.*

The paper is organized as follows. In Section 2, after some necessary notations, in the subsection 2.2 we introduce the family of polynomials obtained considering x_N as an unknown. These polynomials can be expressed in a determinantal form as already seen in [4, 5]. In the present paper we provide a different point of view of the results in [4, 5] and we give some new characterizations. In Section 3 we study a family of iteration functions for computing the roots of these polynomials which turns out to have an underlying symmetric structure which is suitable for handling our family of polynomials. An implementation by Maple - which is one of the most powerful symbolic computational package - of these iteration functions will be also discussed. Finally, in Section 4 we provide some numerical results made with our Maple procedures.

2 Polynomials from generalized Vandermonde determinants

2.1 Notations

First some useful notations and definitions concerning partitions. A *partition* is any (finite or infinite) sequence $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k, \ldots)$ of non-negative integers in decreasing order, that is $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k \geq \cdots$. The non-zero λ_i are called the *parts* of λ. Their number is the *length* of λ and will be denoted by $\ell(\lambda)$.

Definition 2.1 *Given a partition $\lambda = (\lambda_1, \ldots, \lambda_n)$, with $\ell(\lambda) \leq n$, the Schur function s_λ is the symmetric function defined as the quotient*

$$s_\lambda(x_1, \ldots, x_n) = \frac{det\ (x_i^{\lambda_j+n-j})}{det\ (x_i^{n-j})}, \quad 1 \leq i, j \leq n. \tag{2}$$

Since s_λ is the quotient of two homogeneous skew-symmetric polynomials, then is a homogeneous symmetric polynomial of degree $|\lambda|$, where $|\lambda| = \sum_{i=1}^n \lambda_i$, the sum of the parts of λ, is termed the *weight* of λ (cf. [12, p. 40]). The denominator is nothing else than $VDM(x_1, \ldots, x_n)$. Thus, letting $\alpha_j = \lambda_j + n - j$, $\mathbf{x} = (x_1, \ldots, x_n)$ and $V_\alpha(\mathbf{x}) = \det\ (x_i^{\alpha_j})$ we immediately get

$$V_\alpha(\mathbf{x}) = VDM(\mathbf{x})\ s_\lambda\ . \tag{3}$$

That is, every GVD is the product of a classical Vandermonde determinant and a Schur function. As a remark, we remind that Schur functions have many applications and provide a useful tool to study and to compute generalized Vandermonde determinants (cf. e.g. [12, 4, 7, 10]).

Definition 2.2 *Given a partition λ, its* **diagram** *is the set of points $(i,j) \in \mathbb{Z}^2$ such that $1 \le j \le \lambda_i$. The* **conjugate** *of a partition λ, is the partition λ' whose diagram is the transpose of the diagram of λ obtained by reflection in the main diagonal.*

For example, if $\lambda = (4,2,1,0)$, then $\lambda' = (3,2,1,1)$.

Of importance is the **Jacobi-Trudi identity** for Schur functions (cf. [12, formula (3.4), p. 41]):

$$s_\lambda = \det(h_{\lambda_i - i + j}) \ , \qquad 1 \le i,j \le n \ , \tag{4}$$

where the h_p are the *complete elementary symmetric* functions of the variables involved. From Definition 2.2 we get an equivalent form of the Jacobi-Trudi identity. That is, for any partition λ of n,

$$s_\lambda = \det\left(e_{\lambda'_i - i + j}\right)_{1 \le i,j \le n} \ , \tag{5}$$

with λ' the conjugate partition and the e_p representing the *elementary symmetric functions*. Notice that it is commonly assumed that $e_p = 0$ for $p < 0$ and $p > n$.

Taking the sequence of non-negative integers, $0 \le \mu_1 < \mu_2 < \ldots < \mu_n$, and a set of n distinct points, $x_1, \ldots, x_n$, in the determinant

$$V_{\mu;n}(\mathbf{x}) = \left|x_i^{\mu_j}\right|_{1 \le i,j \le n} \ , \tag{6}$$

where the subindex n recall the length of $\mathbf{x}$, then collecting the common factors it can be rewritten as

$$V_{m_{n-1};n}(\mathbf{x}) = \kappa \left|x_i^{m_j}\right|_{1 \le i,j \le n} \ , \tag{7}$$

where $\kappa = \displaystyle\prod_{i=1}^{n} x_i^{\mu_1}$ and $m_j = \mu_j - \mu_1, \ j = 1, \ldots, n$.

2.2 The family of polynomials arising from GVD

From the identity (7), we may consider *univariate polynomials* $Q(x) := V_{m_{n-1};n-1}(x)$ obtained by taking $x = x_n$. The polynomial $Q(x)$ has the obvious determinantal form

$$Q(x) = \begin{vmatrix} 1 & x_1^{m_1} & \ldots & x_1^{m_{n-1}} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n-1}^{m_1} & \ldots & x_{n-1}^{m_{n-1}} \\ 1 & x^{m_1} & \ldots & x^{m_{n-1}} \end{vmatrix} . \tag{8}$$

The next theorem, whose proof could be found in [4, Th. 2.2], provides a interesting factorization of $Q(x)$.

Theorem 2.1 *Let $\alpha = (m_{n-1}, m_{n-2}, \ldots, m_1, 0)$ and $\delta = (n-1, n-2, \ldots, 1, 0)$ be the partitions of length n associated to the determinant (8). Let $\lambda = \alpha - \delta$ and λ' be its conjugate partition. Then,*

$$Q(x) = VDM(x_1, \ldots, x_{n-1}) \left(\prod_{i=1}^{n-1} (x - x_i) \right) P_M(x) \; ; \tag{9}$$

where $M = \lambda_1$. If $\lambda_1 \geq n$ then

$$P_M(x) = \left| \begin{array}{c|cccc} (-1)^{\lambda'_1 - 1} e_{\lambda'_1 - 1} & (-1)^{\lambda'_1} e_{\lambda'_1} & \cdots & (-1)^{\lambda'_1 + M - 1} e_{\lambda'_1 + M - 1} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ (-1)^{\lambda'_n - M} e_{\lambda'_n - M} & (-1)^{\lambda'_n - M + 1} e_{\lambda'_n - M + 1} & \cdots & (-1)^{\lambda'_n} e_{\lambda'_n} \\ ----- & ----- & ----- & ----- \\ 1 & x & \cdots & x^M \end{array} \right| ; \tag{10}$$

Moreover, if $\lambda_1 < n$ then λ' has $n - \lambda_1$ zeros.

Example 1 Let us consider the polynomial:

$$Q(x) := V_{4;2}(x) = \begin{vmatrix} 1 & x_1 & x_1^4 \\ 1 & x_2 & x_2^4 \\ 1 & x & x^4 \end{vmatrix} .$$

By the Theorem 2.1, $V_{4;2}(x) = VDM(x_1, x_2)\,(x - x_1)(x - x_2)\, P_2(x)$. Since $\alpha = (4, 1, 0)$, $\delta = (2, 1, 0)$, $\lambda = (2, 0, 0)$ and $\lambda' = (1, 1, 0)$, then the determinantal form for $P_2(x)$ is:

$$P_2(x) = \begin{vmatrix} e_0 & -e_1 & e_2 \\ -e_{-1} & e_0 & -e_1 \\ 1 & x & x^2 \end{vmatrix}$$

or equivalently

$$P_2(x) = \begin{vmatrix} 1 & -(x_1 + x_2) & x_1 x_2 \\ 0 & 1 & -(x_1 + x_2) \\ 1 & x & x^2 \end{vmatrix} = x^2 + x(x_1 + x_2) + x_1^2 + x_2^2 + x_1 x_2 .$$

Notice that in this case the polynomial is monic and the constant is h_2. This is indeed a general general situation: when $\lambda = (M, 0^{(n-1)})$ (where $0^{(k)}$ says that 0 appears k times in the sequence), $P_M(x)$ is monic with constant term corresponding to h_M (cf. [5, Th. 2.2]). Notice that since $\lambda_1 = 2 < 3 = n$ then λ' has some zeros, as required by the Theorem 2.1.

Remarks.

- Since $Q(x)$ has the determinantal form (10), the points x_i required, must be exactly $n-1$.

- The matrix in (10) shows an interesting property coming from its construction: the constant term of $P_M(x)$ is the $M \times M$ upper-right minor which corresponds to the Schur function s_λ for the points $x_1, \dots, x_{M-1}, 0$ and the same sequence λ. In fact, by using the Jacobi-Trudi identity (5) the Schur function for the sequence λ of length M and the points $x_1, \dots, x_{M-1}, 0$ is $\det\left(e_{\lambda'_i - i + j}\right)_{1 \le i,j \le M}$ with the usual property that $e_p = 0,\ p < 0,\ p > M$, which is the upper-right minor.

- Theorem 2.1 obviously states that in order to know the factorization of $Q(x)$, one has to find the coefficients of $P_M(x)$. In [4] we presented an algorithm for computing the coefficients of $P_M(x)$ based on the Theorem 2.1 and that algorithm can easily implemented by algebraic-symbolic manipulation packages. An implementation of this algorithm, by using *Maple*©, is listed in the Appendix. The Maple procedure `pmx(m,e,b,l,M)` computes firstly the matrix whose determinant is $P_M(x)$ and then the coefficients of $P_M(x)$. The required inputs are: `m` is the number of components in `e`; `e` is the array of the points, i.e. $[x_1, \dots, x_{M-1}]$; `b` the array of the monomial basis $x^i,\ i = 1, \dots, M$ (the procedure does not require x^0!); `l` is the array of the sequence λ' and `M` is the degree of the polynomial, $P_M(x)$ and `x` the independent variable. This is the procedure used for computing all the polynomials in the examples presented in the paper.

It is clear that the form of the sequence α, is fundamental in understanding the form of $P_M(x)$. In the next two claims we show that special forms of α, and consequently λ or λ', give special forms for $P_M(x)$.

Corollary 2.1 *Letting* $m_{n-i} = n - i,\ \ i = 2, \dots, n-2,\ m_{n-1} = m_{n-2} + M + 1 = M + (n-1)$ *for some positive* M *and* $\alpha = (m_{n-1}, m_{n-2}, \dots, m_1, 0)$. *Then* $\lambda = (M, 0^{(n-1)})$ *and* $P_M(x)$ *is a monic polynomial of exact degree* M.

Proof. See [4]. □

Corollary 2.2 *If* $\lambda' = (1^{(M)}, 0^{(n-1)})$ *then* $P_M(x)$ *is monic.*

Proof. Since the conjugate partition of λ' is again λ, hence the conjugate of $\lambda' = (1^{(M)}, 0^{(n-M)})$, is $\lambda = (M, 0^{(n-1)})$ and by the previous theorem we conclude. □

Example 2 This example is a special instance of the third example cited in §1. Let us take the basis of bivariate polynomials of degree $n \ge 3$ restricted to the curve $y = x^3$. The resulting space consists of univariate polynomials of degree $3n$ where the only *missing power* is $3n-1$, i.e. x^{3n-1} is not in the basis, and its dimension is $3n$. Since, $\lambda = \lambda' = (1, 0^{(3n-1)})$ then $M = 1$. The corresponding polynomial $P_1(x) = x - h_1^{(3n-1)} = x - e_1^{(3n-1)}$ is indeed monic of degree 1.

The next Corollary states when a monic $P_M(x)$ reduces to the constant 1.

Corollary 2.3 *Letting* $\alpha = (n-1, n-2, \ldots, 1, 0)$, *so that* $\lambda = (0^{(n)})$. *Then* $P_M(x)$ *is the constant 1.*

Proof. From Theorem 2.1 the polynomial has degree $M = \lambda_1 = 0$ and it is monic, hence $P_M \equiv 1$. $\square$

3 On the computation of the roots of $P_M(x)$

From the fundamental Theorem of Algebra and some of its consequences, we know that every polynomial $p(x)$ of degree $n > 0$ has at least one zero. Moreover, every polynomial $p(x)$ of degree $n > 0$ can be expressed as the product of n linear factors. Hence, $p(x)$ has exactly n zeros not necessarily distinct. Imaginary zeros of polynomials with real coefficients, if they exist, occur in conjugate pairs and a polynomial of odd degree with real coefficients always has at least one real zero. Finally, every polynomial has an associated *discriminant*, which is a polynomial function of its coefficients, that only discriminates the case of a multiple roots. But as is well-known, polynomial equations of degree higher than 4, have no algebraic solution and in this case iteration functions should be found in order to approximate the roots.

In [11], the authors studied a family of high order methods for finding roots of polynomial referred as *Basic family* (which goes back to *Schröder's formulas* dated 1870) and derived an interesting connection with symmetric functions. Attracted by that paper, we applied these formulas to find the real roots of $P_M(x)$ and we also investigated on some connections with our previous results. First we need to recall some notation given in [11].

Hence, for a given polynomial $p(x)$ of degree n, define $d_0(x) = 1$ and for $m \geq 1$, $m \in \mathbb{N}$, let us consider the determinant of the following upper Hessenberg matrix

$$d_m(x) = \begin{vmatrix} p'(x) & \frac{p''(x)}{2} & \cdots & \frac{p^{(m-1)}(x)}{(m-1)!} & \frac{p^{(m)}(x)}{m!} \\ p(x) & p'(x) & \ddots & \ddots & \frac{p^{(m-1)}(x)}{(m-1)!} \\ 0 & p(x) & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \frac{p''(x)}{2} \\ 0 & 0 & \cdots & p(x) & p'(x) \end{vmatrix} . \tag{11}$$

Then, for each $m \geq 2$, the rational function

$$b_m(x) = x - p(x)\frac{d_{m-2}(x)}{d_{m-1}(x)} , \tag{12}$$

defines an iterative algorithm $x_{k+1} = b_m(x_k)$, $k = 0, 1, 2, \ldots$. Moreover, starting from an appropriate initial guess x_0, the sequence $\{x_k\}_{k\geq 0}$ converges with order m to simple roots of $p(x)$ (cf. e.g. [11]).

The determinant $d_m(x)$ can be written also in terms of *complete elementary functions* as follows. Let z_i, $i = 1, \ldots, n$ be the n complex roots of the polynomial $p(x)$ of degree n and define $r_i = 1/(x - z_i)$, $i = 1, \ldots, n$. Then, by means of [11, Lemma 3.1]

$$\frac{p^{(k)}(x)}{k!} = p(x) e_k(r_1, \ldots, r_n), \; 1 \le k \le n \,, \tag{13}$$

where e_k is the kth elementary symmetric function. By [11, Theorem 3.2], $d_m(x)$ can be also rewritten in terms of the mth complete elementary function

$$d_m(x) = p(x)^m h_m(r_1, \ldots, r_n) \,. \tag{14}$$

Hence, by substituting (14) in (12), we get also the connection between the b_m and the complete symmetric functions which shows the symmetric structure of the Basic family:

$$b_m(x) = x - \frac{h_{m-2}(r_1, \ldots, r_n)}{h_{m-1}(r_1, \ldots, r_n)} \,. \tag{15}$$

For the complete elementary functions, which are particular Schur functions, we know that $h_p(r_1, \ldots, r_n) := s_{(p, 0^{(n-1)})}(r_1, \ldots, r_n)$. Then, the ratio

$$\frac{h_{m-1}(r_1, \ldots, r_n)}{h_{m-2}(r_1, \ldots, r_n)}$$

in (15), can be expressed by using the factorization given in the Theorem 2.1. That is, letting $\alpha = (p, 0^{(n-1)}) + (n-1, \ldots, 1, 0)$, then $h_p(r_1, \ldots, r_n) \cdot VDM(r_1, \ldots, r_n) = V_\alpha$. Moreover, recalling that (cf. [4, p. 278, formula (19)])

$$h_p = \det\left(e_{\lambda^* - i + j}\right), \;\; 1 \le i, j \le n \,,$$

where $\lambda^* = (1^{(n)})$, we can express h_p as a determinant of an upper Hessenberg matrix whose elements are the elementary symmetric functions e_j.

In conclusion, we have proved that h_{m-2} and h_{m-1} are indeed related to the determinant (11) through the equalities (13).

4 Numerical experiments

By assumption, the distinct points $x_1, \ldots, x_{n-1}$ belonging to some interval $[a, b]$ are the roots of the polynomial $Q(x) = V_{m_{n-1}; n-1}(x)$. We present here some experiments we have done by using the Maple procedures we have implemented (see the Appendix). Supported by many numerical experiments, we observed that the roots of $P_M(x)$ seems to distribute as follows.

Conjecture

1. If M is even, they are pairwise complex.

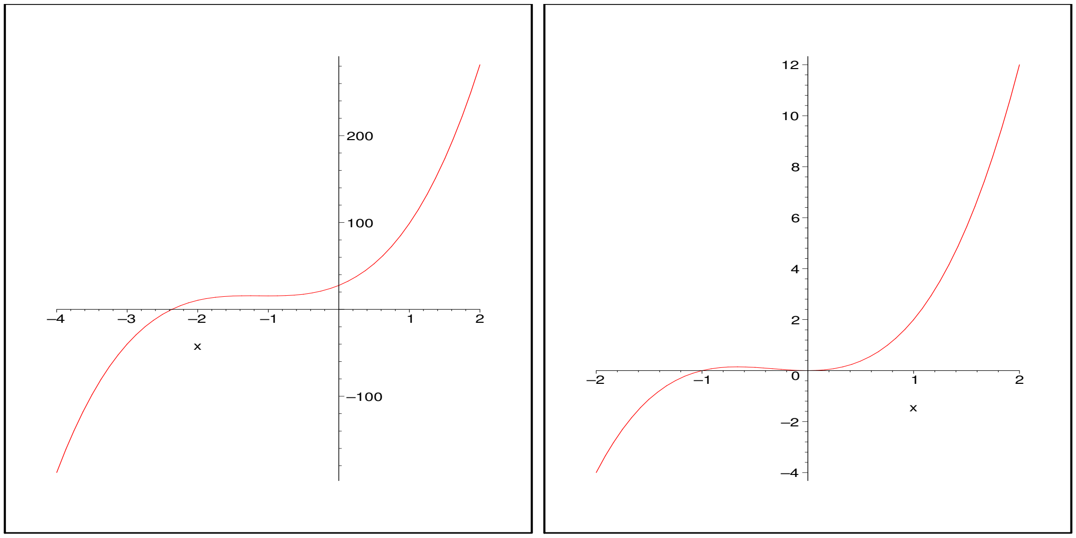

Figure 1: *Left*: $P_3(x)$ when $x_1 = 1.5,\ x_2 = 1.9$. *Right*: $P_3(x)$ when $x_1 = 0,\ x_2 = 1$ for Example 3.

2. If M is odd, there exist $(M-1)/2$ pairwise complex conjugate roots and one real ξ not belonging to $[\min\{x_i\}, \max\{x_i\}]$.

In both cases the real roots do not (obviously) coincide with the points $x_i, \quad i = 1, \ldots, n-1$.

Remarks.

- When M is even the argument is obvious. In fact, the x_i are exactly the *real* roots of $Q(x) = V_{m;n}(x)$ in [a,b] (which all do not appear in the polynomial $P_M(x)$). So, the only roots of $P_M(x)$ should be pairwise complex. Moreover, the factorization induced by our results, for *even* $M \geq 2$, can be seen as a constructive way for generating a family of polynomials having (only) complex roots.
- When M is odd, we have only verified numerically that distribution. The only common situation is that at least one root is outside the interval $[\min\{x_i\}, \max\{x_i\}]$.

Example 3 Letting $\alpha = (5, 3, 0)$ so that $\lambda = (3, 2, 0)$ and $\lambda' = (2, 2, 1)$ and suppose $x_i \in [0, 2]$. Thus, from the Theorem 2.1

$$P_3(x) = \begin{vmatrix} -e_1 & e_2 & 0 & 0 \\ 1 & -e_1 & e_2 & 0 \\ 0 & 0 & 1 & -e_1 \\ 1 & x & x^2 & x^3 \end{vmatrix}.$$

Since the points x_i lye on $[0, 2]$, the roots of $P_3(x)$ should be one real not belonging to $[0, 2]$ and two conjugate complex. We tried many different choices of x_1, x_2 and in all cases the conjecture was verified.

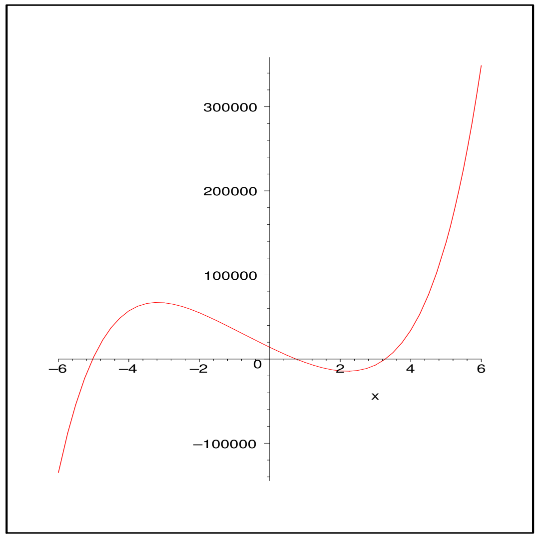

Figure 2: The polynomial $P_5(x)$ of Example 4.

In Figure 1 we show the plots of $P_3(x)$ for two different choices of $x_1, x_2 \in [0, 2]$. The corresponding roots are: $-2.3681, -0.5159 \pm i\,1.0357$ and $-1, 0, 0$, respectively.

Another example for a bigger M.

Example 4 Suppose $\alpha = (10, 5, 4, 1, 0)$ so that $\lambda = (6, 2, 2, 0, 0)$ and $\lambda' = (3, 3, 1, 1, 1, 1)$. In Table 1 we summarize some choices of $x_i,\ i = 1, 2, 3, 4$ and the roots corresponding to $P_6(x)$. Notice that in this example the interval $[a, b]$ corresponds to the interval $[\min\{|x_i|\}, \max\{|x_i|\}]$.

If $\alpha = (9, 5, 3, 1, 0)$, so that $\lambda' = (3, 2, 1, 1, 1)$, when $x_1 = -3,\ x_2 = -2,\ x_3 = 1,\ x_4 = 5$, the roots of $P_5(x)$ in $[-3, 5]$ are $0.75, 3.28$ plus two complex conjugate that is $-0.001 \pm i\,6.15$. We observe that there is another root in -5.026 which is outside the interval $[-3, 5]$ delimited by the points x_i (see Figure 2).

points	roots
$x_i = i,\ i = 1, \ldots, 4$	$1.0567 \pm i\,5.3969$ $-4.8931 \pm i\,3.2355$ $-1.1656 \pm i\,1.3515$
$x_1 = -2, x_2 = -1$ $x_3 = 1, x_4 = 2$	$\pm 1.0299 \pm i\,1.7515$ $0. \pm i\,0.9928$
$x_1 = -\frac{2}{3}, x_2 = -\frac{1}{2}$ $x_3 = 0, x_4 = \frac{1}{8}$	$0.7521 \pm i\,0.5032$ $-0.1744 \pm i\,0.8276$ $-0.0569 \pm i\,0.1273$

Table 1: The roots of $P_6(x)$ for different choices of the x_i's for Example 4.

In Tables 2 and 3, we show the behavior of the Newton's method and the Basic family for the computation of the real roots of the polynomials P_3 and P_5 discussed in the previous two examples. In both Tables, the results presented have been obtained by running the two iterative methods and using as stopping criterion the test on the relative error between two approximations with a tolerance of 10^{-8}. In Table 2, the root 0.0 is double and in order to preserve the order of convergence of the two methods, we applied b_m to $p(x)/p'(x)$, instead of $p(x)$, whose roots are all simple and the same as those of $p(x)$.

Initial guess	root	Newton	Basic family
$x_0 = -1.0$	-2.368	50	11
$x_0 = -0.8$	-1.0	5	3
$x_0 = 1.0$	0.0	5	4

Table 2: Number of iterations for the Newton's method and the Basic family for computing the real roots of $P_3(x) = 8.71x^3 + 29.614x^2 + 32.946x + 27.6165$ which corresponds to the choice $x_1 = 1.5$ and $x_2 = 1.9$; and $P_3(x) = x^3 + x^2$ which corresponds to the choice $x_1 = 0.$ and $x_2 = 1.0$.

Initial guess	root	Newton	Basic family
$x_0 = 0.5$	0.752	3	2
$x_0 = 3.0$	3.277	4	3
$x_0 = -4.0$	-5.026	6	2

Table 3: Number of iterations for the Newton's method and the Basic family for computing the real roots of $P_5(x) = 30x^5 + 30x^4 + 600x^3 + 1500x^2 - 20160x + 14040$, which corresponds to the choice $x_1 = -3$, $x_2 = -2$, $x_3 = -1$, $x_4 = 5$.

Final remarks. From the Example 4, we observe that a small change in the components of α, makes a strong *influence* on the distribution of the roots of P_M. This suggests that the conditioning of the problem of finding the roots of P_M depends also on the sequence α. As a second observation, we point out that when all the x_i are *integers*, the coefficients of $P_M(x)$ are integers numbers growing in size with M (i.e. they have a lot of digits). In this latter case, a suitable software for computing these roots is MPSolve 2.0 (cf. [2]).

Acknowledgments. This work has been supported by the ex 60% funds of the University of Verona, year 2005 and the MIUR project Prin 2003 "Modelli Numerici per applicazioni avanzate in meccanica dei fluidi ed elettromagnetismo".

References

[1] R. Abu-Saris and W. Ahmad. Generalized Exponential Vandermonde Determinant and Hermite Multi-point Discrete Boundary Value Problem, *SIAM J. Matrix Anal. Appl.*, **25(4)**, 921–929 (2004).

[2] D. A. Bini and G. Fiorentino. MPSolve: Numerical Computation of Polynomials Roots: v 2.0. FRISCO report 1998, available at http://www.dm.unipi.it/pages/bini/public_html/ric.html.

[3] L. Bos and S. De Marchi. Fekete points for bivariate polynomials restricted to $y = x^m$, *East J. Approx.* **6(2)**, 189–200 (2000).

[4] S. De Marchi. Polynomials arising in factoring Generalized Vandermonde determinants: an algorithms for computing their coefficients, *Math. Comput. Modelling*, 34, 271–281 (2001).

[5] S. De Marchi. Polynomials arising in factoring Generalized Vandermonde determinants II: an condition for monicity, *Appl. Math. Lett.*, **15(5)**, 627–632 (2002).

[6] S. De Marchi and C. Roveredo. On blossoming in integer Müntz spaces. *Int. Math. J.*, **5(1)**, 61-66 (2004).

[7] T. Ernst. Generalized Vandermonde determinants, U.U.D.M Tech. Report nr. 6 (2000),`www.math.uu.se/~thomas/`.

[8] R. P. Flowe and G. A. Harris. A note on generalized Vandermonde determinants, *SIAM J. Matrix Anal. Appl.*, 14(4), 1146–1151 (1993).

[9] E. R. Heineman. Generalized Vandermonde determinants, *Proc. Royal Society Edimburgh*, 464–476 (1929).

[10] R. C. King. Generalized Vandermonde determinants and Schur functions, *Proc. AMS*, 48(1), 53–56 (1975).

[11] Y. Jin and B. Kalantari. Symmetric functions and root-finding algorithms, Adv. in Appl. Math. 34, 156–174 (2005).

[12] I. G. Macdonald. Symmetric functions and Hall polynomials, Oxford Univ. Press, Oxford OX2 6DP (England), Second Ed. (1995).

[13] S. Martucci and B. Paternoster. Vandermonde-type matrices in two step collocation methods for special second order ordinary differential equations. In *Computational Science ICCS2004*, Lecture Notes in Computer Science 3039, Part IV, pp. 416–425, Springer-Verlag, 2004.

[14] S. Yang, H. Wu, Q. Zhang. Generalization of Vandermonde determinants, *Lin. Alg. Appl.*, 36, 201–204 (2001).

5 Appendix

Our first Maple procedure, builds the determinant (10). Inputs: `m` is the number of components in `e`; `e` is the array of the points, i.e. $[x_1, \ldots, x_{M-1}]$; `b` the array of the monomial basis x^i, $i = 1, \ldots, M$ (the procedure does not require x^0!); `l` is the array of the sequence λ' and `M` is the degree of the polynomial, $P_M(x)$ and `x` the independent variable.

```
pmx:=proc(m,e,b,l,M,x)
local a,i,p,x,j,c1,s;
```

```
global dd,c;

description "Built the determinant corresponding to the polynomial P_M(x)":

with(linalg);

a:=array(1..M+1,1..M+1);
p:=1;
for i from 1 to m do
 p:=p*(e[i]-x);
od:

c1:=collect(expand(p,b),x);
 with(PolynomialTools);
 c:=CoefficientVector(c1,x);

for i from 1 to M do
 for j from 1 to M+1 do
  s:=l[i]-i+j-1;
  if (s=0) then a[i,j]:=1;
  elif (s>m) or (s<0) then
  a[i,j]:=0;
  else
 a[i,j]:=c[m-s+1];
end if
od:
od:

for i from 1 to M+1 do
 a[M+1, i]:=x^(i-1);
od;

description "Now we print, for a check, the determinant and its factorization":
print(det(a));
dd:=convert(collect(expand(det(a)),x),polynom); end;
```

The second Maple procedure, that we call `dm`, builds the determinant D_m required in the construction of the Schröder's family of formulas for computing zeros of polynomials. Inputs: `pol` is the polynomial, `m` is the order up to which we wish to compute `dm`, $0 \leq m \leq$ `degree`(`pol`).

```
dm:=proc(pol,m)
local dd,d1,k,i,j,s;
 global dm,c,x;

description "here we build the determinant that implements the
Shroeder's family of formulas for
computing zeros of polynomials;
pol is the polynomial,
```

```
m is the order up to which we wish to compute
dm (notice that m can be degree(pol))":

with(linalg): dd:=array(1..m,1..m);
if (m=0) then
  d1:=1;
elif (m=1) then
  d1:=diff(pol,x);
else
 for i from 1 to m do
 for j from 1 to m do
   s:=j-i+1;
   if (s=0) then
      dd[i,j]:=pol;
   elif (s<0) then
      dd[i,j]:=0;
   else
      dd[i,j]:=diff(pol,x$s)/s!;
   end if;
 end do;
end do;
d1:=det(dd);
 end if;
 end proc;
```

The third Maple procedure, that we call `itera`, simply implements the Schröder's iteration or the Basic family. Inputs: `p`, the polynomial and `xstart` the initial guess.

```
itera:=proc(p,xstart)
description "this is the Schroeder's iteration"
local x0,x1,k;
 global x,y,dm;
 x0:=xstart;
 x1:=eval(x-p*dm(p,degree(p)-2)/dm(p,degree(p)-1),x=x0);
 k:=1;
 while (abs(x1-x0)>10^(-8)*abs(x0)) and (k <= 100) do
  x0:=x1;
  x1:=eval(x-p*dm(p,degree(p)-2)/dm(p,degree(p)-1),x=x0);
  k:=k+1;
 end do;
 y:=x1;
 printf("Iterations = %g \n",k-1);
 printf("x1 = %f \n",x1);
 end proc;
```

Remarks. The Newton's method is obtained by using `itera` and setting `degree(p)=2`. In the case of multiple roots, we applied both methods to $p(x)/p'(x)$ instead of $p(x)$ in order to preserve the convergence order.