BULLETIN OF THE
GREEK MATHEMATICAL SOCIETY
Volume 54, 2007 (221–237)

Computations for Minors of Hadamard Matrices

Christos Kravvaritis and Marilena Mitrouli

Received 12 December 2006 Accepted 2 September 2007

Abstract

In the present paper we are interested in calculating values of minors for Hadamard matrices. By taking advantage of the special properties of Hadamard matrices, we aim at deriving analytical formulas for the $(n-j)\times(n-j)$ minors of them, $j \geq 1$. For our purpose we carry out the appropriate determinant evaluations and develop in this way a theoretical technique that provides general results up to minors of order $n-3$.

The benefit of the proposed theoretical technique is that it can be adopted for developing a computer algorithm for the same purpose. So, the obstacles originating from the computations by hand can be eliminated by implementing the initial notion in a computer algorithm. Theoretically, the proposed algorithm can work for every values of n and j. We present the results of its application for $n = 12,\ 16$ and $j = 1, \ldots, 7$. So, we give a new way for calculating the $n-j$ minors of Hadamard matrices.

Key Words and Phrases: Determinants, Minors, Hadamard matrices, symbolic computations.

AMS Subject Classification: 15A15; 05B20; 65F40; 65F05; 65G50;

1. Introduction

Determinants are an old and intensively studied mathematical object, but they are even nowadays of great research interest, c.f. e.g. [9], [5]. This happens because, although there have been found methods for calculating the determinant of every possible matrix, it is always interesting and useful to find formulas of determinants of matrices with special structure and properties. Such formulas have already been demonstrated for Vandermonde matrices, Hankel matrices and Cauchy matrices. The benefit of such an effort is the fact that analytical formulas provide a more efficient evaluation of determinants and also the possibility to offer more insight on some properties of a matrix. On the contrary, evaluations using traditional expansion methods

*This research was financially supported by PENED 03ED 740 of the Greek General Secretariat for Research and Technology

for determinants might lead to an increase of the computational cost and finally to a computational failure. Moreover, there can be developed improved techniques for computing determinants if the elements of the matrices can have only some restricted values in some sense, e.g. integer matrices (involving modulo computations) or matrix polynomials [4]. In our research we focus on calculating principal minors of Hadamard matrices.

An $(1, -1)$ matrix H of order n satisfying $HH^T = H^T H = nI_n$ is called a *Hadamard matrix of order* n or simply a *Hadamard matrix*. It can be proved that always $n \equiv 0\,(mod 4)$. An important property of the Hadamard matrices, which follows directly from the definition, is that every two distinct rows and columns of it are orthogonal to each other, which means that their inner product is zero.

The motivation for studying the minors of Hadamard matrices lies in the fact that the magnitude of the pivots appearing after the application of GE (Gaussian Elimination) operations on a Completely Pivoted (CP, no exchanges are needed during GE with complete pivoting) matrix W is given by

$$p_j = \frac{W(j)}{W(j-1)}, \quad j = 1, 2, \ldots, n, \quad W(0) = 1, \tag{1}$$

where the denotation $W(j)$ is declared in paragraph Notation in Section 2.

This result was proved in [3]. So, it is obvious that the calculation of minors is important in order to study pivot structures of CP Hadamard matrices and moreover, the growth conjecture associated with them, which was formulated by Cryer [2]. In this study we intend to calculate $(n - j)$ minors of a Hadamard matrix in general, independent of the CP property.

Initially, we aimed at calculating by hand minors of Hadamard matrices theoretically. We derive analytical formulas for the $n-1$, $n-2$ and $n-3$ minors of an $n \times n$ Hadamard matrix and obtain the values $n^{(n/2)-1}$, 0 or $2n^{(n/2)-2}$ and 0 or $4n^{(n/2)-3}$, respectively. The proofs for these results were carried out with appropriate determinant manipulations and by taking always into account the special properties of these matrices. A brief examination of the proof for the $n - 3$ minors indicates that the calculations, which should be done by hand for the $n - 4$, $n - 5$ etc. minors, can be done similarly, but they will be even more strenuous and complicated.

Nevertheless, a more careful examination of the proof for the $n - 3$ minors with more insight of the calculations taking place, reveals that we can carry out all the appearing algebraic calculations in an algorithmic sense and implement them in a Symbolic Computation software, such a Maple.

2. Theoretical calculations

Preliminary Results. 1. Let

$$A = (k - \lambda)I_v + \lambda J_v,$$

where $k,\ \lambda$ are integers. Then,

$$det\ A = [k + (v-1)\lambda](k-\lambda)^{v-1} \tag{2}$$

and

$$A^{-1} = \frac{1}{k^2 + (v-2)k\lambda - (v-1)\lambda^2}\{[k + (v-2)\lambda + \lambda]I - \lambda J\} \tag{3}$$

2. Let $B = \begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix}$. Then

$$det\ B = det\ B_1 \cdot det(B_4 - B_3 B_1^{-1} B_2) \tag{4}$$

Notation. Throughout this paper we assume, without loss of generality, that the entries of the first row and column of a Hadamard matrix are always $+1$ (normalized form), because this can be achieved easily by multiplying columns and/or rows with -1 and leaves unaffected the magnitude of the determinant. The elements of an $(1,-1)$ matrix will be denoted by $(+,-)$. I_n and J_n stand for the identity matrix of order n and the matrix with ones of order n, respectively. $J_{m,n}$ is the matrix with ones of order $m \times n$. We write $A(j)$ for the absolute value of the determinant of the $j \times j$ principal submatrix in the upper left corner of the matrix A. M_j denotes all possible absolute values of the $j \times j$ minors of a matrix. We denote with $x_{m\times n}$ the $m \times n$ block with elements x, x real, and with $X_{m\times n}$ the $m \times n$ block with the specific form of the matrix X.

We write $J_{b_1,b_2,\cdots,b_z}$ for the all ones matrix with diagonal blocks of sizes $b_1 \times b_1,\ b_2 \times b_2\ \cdots\ b_z \times b_z$, and $a_{ij}J_{b_1,b_2,\cdots,b_z}$ for the matrix, for which the elements of the block with corners $(i + b_1 + b_2 + \cdots + b_{j-1}, i + b_1 + b_2 + \cdots + b_{i-1})$, $(i + b_1 + b_2 + \cdots + b_{j-1}, b_1 + b_2 + \cdots + b_i)$, $(b_1 + b_2 + \cdots + b_j, i + b_1 + b_2 + \cdots + b_{i-1})$, $(b_1 + b_2 + \cdots + b_j, b_1 + b_2 + \cdots + b_i)$ are a_{ij} an integer.

We write $(k_i - a_{ii})I_{b_1,b_2,\cdots,b_z}$ for the matrix direct sum $(k_1 - a_{11})I_{b_1} + (k_2 - a_{22})I_{b_2} + \cdots + (k_z - a_{zz})I_{b_z}$.

We introduce also another important notion for our work. Two matrices are said to be *Hadamard equivalent* or *H-equivalent* if one can be obtained from the other by a sequence of the operations:

1 interchange any pairs of rows and/or columns;

2 multiply any rows and/or columns through by -1.

Proposition 2.1 *Let H be a Hadamard matrix of order n. Then all possible $(n-1) \times (n-1)$ minors of H are: $M_{n-1} = n^{\frac{n}{2}-1}$.*

Proof. Since H be a Hadamard matrix of order n, we suppose that it can written in the following form:

$$H = \begin{bmatrix} + & + & \dots & + \\ + & & & \\ \vdots & & \text{B} & \\ + & & & \end{bmatrix}.$$

From the definition of H, $HH^T = nI_n$, follows that the $(n-1)\times(n-1)$ matrix BB^T has the form

$$BB^T = nI_{n-1} - J_{n-1}.$$

Then, from (2), we have
$det\ BB^T = [n-1-(n-1-1)](n-1+1)^{n-1-1} = n^{n-2}$. So $det\ B = n^{\frac{n}{2}-1}$.
From this proof it becomes obvious that we obtain the same value for $det\, B$, independently of its possible position inside H. Hence, we can conclude that $M_{n-1} = n^{\frac{n}{2}-1}$.

Proposition 2.2 *Let H be a Hadamard matrix of order n. Then all possible $(n-2)\times(n-2)$ minors of H are: $M_{n-2} = 0$ or $2n^{\frac{n}{2}-2}$.*

Proof. There are two possible cases, up to H-equivalence, for the upper left corner:

$$\begin{bmatrix} + & + \\ + & - \end{bmatrix} \text{ and } \begin{bmatrix} + & + \\ + & + \end{bmatrix}.$$

We will carry out the proof for the first case, since the second case can be handled similarly.
H can be written in the following form:

$$H = \begin{bmatrix} + & + & \overbrace{+ \ \dots \ +}^{u} & \overbrace{+ \ \dots \ +}^{v} \\ + & - & + \ \dots \ + & - \ \dots \ - \\ + & + & & \\ \vdots & \vdots & & \\ + & + & \text{C} & \\ + & - & & \\ \vdots & \vdots & & \\ + & - & & \end{bmatrix},$$

where the first two columns contain also u $[+,+]$ and u $[+,-]$.
From the order of the matrix H and the orthogonality of its two first rows we get the following two equations

$$\begin{cases} u+v = n-2 \\ u-v = 0 \end{cases},$$

which imply $u = v = \dfrac{n-2}{2}$.
According to the definition of H, the $(n-2)\times(n-2)$ matrix CC^T has the form

$$CC^T = \begin{bmatrix} C_1 & \mathbf{O} \\ \mathbf{O} & C_1 \end{bmatrix}, \text{ with } C_1 = nI_u - 2J_u.$$

According to (4),

$$det\,C_1 = [n-2-2(\tfrac{n-2}{2}-1)](n-2+2)^{\frac{n-2}{2}-1} = n^{\frac{n}{2}-2}.$$

Hence, $det\,CC^T = (det\,C_1)^2 = n^{n-4}$ and $det\,C \equiv M_{n-2} = n^{\frac{n}{2}-2}$.
Similarly, the other case for the upper left 2×2 corner gives the value $M_{n-2} = 0$,

Proposition 2.3 *Let H be an $n\times n$ Hadamard matrix. Then all possible $(n-3)\times(n-3)$ minors of H are: $M_{n-3} = 0$ or $4n^{\frac{n}{2}-3}$.*

Proof. There are 16 possible cases, up to H-equivalence, for the upper left corner:

$$\begin{bmatrix} + & + & + \\ + & \pm & \pm \\ + & \pm & \pm \end{bmatrix}$$

We will illustrate the proof for one matrix, since the other cases can be handled absolutely similarly. Since we have that matrix H is an $n\times n$ Hadamard matrix, let us suppose that it can be written in the following form:

$$W = \left[\begin{array}{ccc|cccc} + & + & + & \overbrace{+ \cdots +}^{u} & \overbrace{+ \cdots +}^{v} & \overbrace{+ \cdots +}^{x} & \overbrace{+ \cdots +}^{y} \\ + & - & - & + \cdots + & + \cdots + & - \cdots - & - \cdots - \\ + & + & - & + \cdots + & - \cdots - & + \cdots + & - \cdots - \\ + & + & + & & & & \\ & \vdots & & & & & \\ + & + & + & & & & \\ + & + & - & & & & \\ & \vdots & & & D & & \\ + & + & - & & & & \\ + & - & + & & & & \\ & \vdots & & & & & \\ + & - & + & & & & \\ + & - & - & & & & \\ & \vdots & & & & & \\ + & - & - & & & & \end{array}\right],$$

where the first three columns contain also u $[+,+,+]$, v $[+,+,-]$, x $[+,-,+]$ and y $[+,-,-]$.
From the order of the matrix H and the orthogonality of its three first rows we get the following system of four equations

$$\begin{cases} u+v+x+y = n-3 \\ u+v-x-y = 1 \\ u-v+x-y = -1 \\ u-v-x+y = -1 \end{cases},$$

which has the exact solution

$$u = x = y = \tfrac{n}{4} - 1$$
$$v = \tfrac{n}{4}.$$

According to the properties of an $n \times n$ Hadamard matrix, the $(n-3) \times (n-3)$ matrix DD^T has the form $DD^T = \begin{bmatrix} E_{u\times u} & F \\ F^T & G \end{bmatrix}$, where E is a matrix of the form $E = nI_u - 3J_u$, $F = \begin{bmatrix} -1_{u\times v} & -1_{u\times x} & 1_{u\times y} \end{bmatrix}$ and $G = \begin{bmatrix} E_{v\times v} & 1_{v\times x} & -1_{v\times y} \\ 1_{x\times v} & E_{x\times x} & -1_{x\times y} \\ -1_{y\times v} & -1_{y\times x} & E_{y\times y} \end{bmatrix}$.
So, according to (4),

$$det\, DD^T = det\ E_{u\times u} \cdot det(G - F^T E_{u\times u}^{-1} F). \tag{5}$$

From (2) we have

$$det\ E_{u\times u} = \frac{(n+12)n^{\frac{n-8}{4}}}{4} \tag{6}$$

and from (3) $E_{u\times u}^{-1} = \frac{n+24}{n(n+12)} I_u - \frac{12}{n(n+12)} J_u$.
Hence,

$$G - F^T E_{u\times u}^{-1} F \quad = \quad \begin{bmatrix} K_{1v\times v} & N_2 \\ N_2^T & N_1 \end{bmatrix},$$

where the appearing blocks $K_{1v\times v}$, N_1 and N_2 are calculated.
So, according to (4),

$$det(G - F^T E_{u\times u}^{-1} F) = det\ K_{1v\times v} \cdot det(N_1 - N_2^T {K_{1v\times v}}^{-1} N_2). \tag{7}$$

From this point and on, the idea of the proof is to apply consecutively formula (4) appropriately for the appearing block matrices and carry out the calculations with help of (2) and (3).

We proceed in an absolute similar way like before in order to calculate $det\ K_{1v\times v}$ and $det(N_1 - N_2^T K_{v\times v}^{-1} N_2)$, by making use of (2), (3) and (4).
We have

$$det\, K_{1v\times v} = \frac{4 \cdot n^{\frac{n}{4}}}{n+12} \tag{8}$$

$$N_1 - N_2^T {K_{1v\times v}}^{-1} N_2 = \begin{bmatrix} P_{1x\times x} & Q_{1x\times y} \\ Q_{1y\times x} & P_{2y\times y} \end{bmatrix},$$

where the blocks $P_{1x\times x}$, $Q_{1x\times y}$ and $P_{2y\times y}$ are obtained appropriately.

According to (4),

$$det(N_1 - N_2^T {K_{1v\times v}}^{-1} N_2) = det\ P_{1x\times x} \cdot det(P_{2y\times y} - Q_{1y\times x} {P_{1x\times x}}^{-1} Q_{1x\times y}). \tag{9}$$

From (2) and (3) we have

$$det\, P_{1x\times x} = 4n^{\frac{n-8}{4}} \tag{10}$$

$$P_{2y\times y} - Q_{1y\times x} {P_{1x\times x}}^{-1} Q_{1x\times y} = R_{3y\times y}, \tag{11}$$

where $R_{3y\times y} = nI_y - 4J_y$.
Equation (2) implies

$$det\, R_{3y\times y} = 4n^{\frac{n-8}{4}} \tag{12}$$

Finally, from (5), (6), (7), (8), (9), (10), (11) and (12) we have

$$det\ DD^T = det\, E_{u\times u}\ det\, K_{1v\times v}\ det\, P_{1x\times x}\ det\, R_{3y\times y} = 16n^{n-6}$$

Hence, $det\ D \equiv M_{n-3} = 4n^{\frac{n}{2}-3}$.
Similarly we handle all possible remaining cases for the upper left and we obtain the results $M_{n-3} = 0, 4n^{\frac{n}{2}-3}$.

We note that, although the calculations done by hand required for the $n-4, n-5$ etc. minors will be more complicated, it is easy to observe that they follow a predictable, standard procedure, which seemed challenging to develop from an algorithmic point of view and finally to implement it on a Computer Algebra Package, such as Maple.

3. An Algorithm to Find Values of Minors of Hadamard Matrices

3.1. *Description of the Algorithm*

We are interested in calculating the $(n-j)\times(n-j)$ minors of an $n\times n$ Hadamard matrix. The proposed method is based on the idea that every Hadamard matrix can be written in the following form

$$H = \begin{bmatrix} M & U_j \\ U_j^T & D \end{bmatrix},$$

where $M,\ D$ are $j\times j$ and $(n-j)\times(n-j)$ matrices, respectively, and U_j is the matrix of order $j\times(n-j)$, which contains all possible 2^{j-1} columns with elements ± 1 starting with $+1$. The value of the desired $(n-j)\times(n-j)$ minor is actually the determinant of D. From the definition of a Hadamard matrix $HH^T = I_n$ follows easily that the elements of H can be permuted by H-equivalent operations, so that the $(n-j)\times(n-j)$ matrix DD^T appears in the form

$$DD^T = (n-j-a_{ii}) I_{u_1,u_2,\cdots,u_{2^{j-1}}} + a_{ik} J_{u_1,u_2,\cdots,u_{2^{j-1}}},$$

where $(a_{ik}) = (-u_i \cdot u_k)$, $a_{ii} = (-u_i \cdot u_i) = -j$, with $\cdot$ the inner product. Then $det\, D$ can be calculated by successive applications of formula (4), in combination with (2) and (3).

This can be achieved by setting initially $E_1,\ F_1,\ F_1^T$ and G_1 the upper left $u_1 \times u_1$, the upper right $u_1 \times (n-j-u_1)$, the lower left $(n-j-u_1) \times u_1$ and the lower right $(n-j-u_1) \times (n-j-u_1)$ submatrix of DD^T, respectively. Next we denote by E_2 the upper left $u_2 \times u_2$, F_2 the upper right $u_2 \times (n-j-u_1-u_2)$, F_2^T the lower left $(n-j-u_1-u_2) \times u_2$ and G_2 the lower right $(n-j-u_1-u_2) \times (n-j-u_1-u_2)$ submatrix of $G_1 - F_1^T E_1^{-1} F_1$, respectively. Then, according to (4), $det\, DD^T = det\, E_1 \cdot det(G_1 - F_1^T E_1^{-1} F_1) = det\, E_1 \cdot det\, E_2 \cdot det(G_2 - F_2^T E_2^{-1} F_2)$. From (2) we calculate $det\, E_2$ and we proceed on calculating $G_2 - F_2^T E_2^{-1} F_2$ with help of (3). After this first iteration, we set E_3 the upper left $u_3 \times u_3$ submatrix of $G_2 - F_2^T E_2^{-1} F_2$, F_3 the upper right $u_3 \times (n-j-u_1-u_2-u_3)$ submatrix of $G_2 - F_2^T E_2^{-1} F_2$ etc. We continue in a similar manner the assignment of the matrices $E_3, \ldots, E_{2^{j-1}-1}$, until all their determinants are calculated. The last matrix $E_{2^{j-1}}$ of this sequence is of the form $(k-\lambda) I_{u_{2^{j-1}}} + \lambda J_{u_{2^{j-1}}}$ and its determinant can be calculated directly with help of (2). Finally, $det\, DD^T = det\, E_1 \cdot det\, E_2 \ldots det\, E_{2^{j-1}-1} \cdot det\, E_{2^{j-1}}$.

We note that for the computation of minors of order $n-j$, $j > 3$, there will exist parameters in the solutions of the familiar system with unknowns the number of columns of U_j, because the number of unknowns is greater than the number of equations. The values of the parameters can be bounded by using Lemma 3.1. So, in these cases it will not be possible to obtain general results for the values of $n-j$ minors, but only for specific n, by letting the parameters to attain all values $0, \ldots, \frac{n}{4}$.

We give the following Lemma 3.1, which is useful for obtaining constraints on the number of columns of a Hadamard matrix, and moreover for limiting the calculations in our algorithm.

Lemma 3.1 *For the first j rows of a normalized Hadamard matrix of order n, $n > 3$, and for all the 2^{j-1} possible columns of U_j, it holds*

$$0 \le u_i \le \frac{n}{4}\ ,\ for\ \ i = 1, \ldots, 2^{j-1}$$

3.2. *The Algorithm*

We provide an Algorithm for calculating all possible $(n-j) \times (n-j)$ minors of a Hadamard matrix of order n.

Algorithm Minors

Step 1: **Read** all $j \times j$ matrices M, which can exist in the upper left corner of a Hadamard matrix $H = \begin{bmatrix} M & U_j \\ U_j^T & D \end{bmatrix}$.

Step 2: **For every** matrix M

Form the system of $1 + \binom{j}{2}$ equations and 2^{j-1} unknowns u_i that results from counting of columns and the inner products of every two distinct rows of

the matrix $[M \;\; U_j]$.

Solve the system for all u_i.

Step 3: **For** all the parameters $u_{p,i} = 0, \ldots, \frac{n}{4}$, $i = 1, \ldots, 2^{j-1} - 1 - \binom{j}{2}$

Step 4: **If** $u_1 \geq 1$ and $u_i \geq 0$, $i = 2, \ldots, 2^{j-1}$ and u_i integers, $i = 1, \ldots, 2^{j-1}$

$$DD^T = (n-j-a_{ii})I_{u_1,u_2,\cdots,u_{2^{j-1}}} + a_{ik}J_{u_1,u_2,\cdots,u_{2^{j-1}}} \equiv \begin{bmatrix} E_1 & F_1 \\ F_1^T & G_1 \end{bmatrix}$$

$$G_1 - F_1^T E_1^{-1} F_1 \equiv \begin{bmatrix} E_2 & F_2 \\ F_2^T & G_2 \end{bmatrix}$$

Else there are no acceptable solutions

End If

Step 5: **For** $k = 2, \ldots, 2^{j-1} - 2$

$$G_k - F_k^T E_k^{-1} F_k \equiv \begin{bmatrix} E_{k+1} & F_{k+1} \\ F_{k+1}^T & G_{k+1} \end{bmatrix}$$

End

Step 6: $E_{2^{j-1}} \equiv G_{2^{j-1}-1} - F_{2^{j-1}-1}^T E_{2^{j-1}-1}^{-1} F_{2^{j-1}-1}$

Step 7: $det\, DD^T := \prod_{i=1}^{2^{j-1}} det\, E_i, \;\; det\, D = \sqrt{det\, DD^T}$

End {for all the parameters u_i }

End {for every matrix M }

End {of Algorithm}

Remarks on the Algorithm

1 In Step 1 the algorithm takes as input all possible $j \times j$ matrices M, which can exist in the upper left corner of a Hadamard matrix. In order to determine whether such a submatrix can or cannot exist inside a Hadamard matrix, we have developed counting techniques in algorithmic form, similarly to the ones introduced in [8]. So, in this way we exclude a significant number of matrices M from the process and consequently, we limit its computational cost.

2 Step 3 corresponds actually to successive "for" loops of the form

For $u_{p,1} = 0, \ldots, \frac{n}{4}$

For $u_{p,2} = 0, \ldots, \frac{n}{4}$

$\vdots$

For $u_{p,z} = 0, \ldots, \frac{n}{4}$,

where $z = 2^{j-1} - 1 - \binom{j}{2}$.

3 In Steps 5 and 6 all matrix multiplications and inversions are not performed explicitly, but in an effective sense, as can be seen in the implementation of the Algorithm in the following Section.

4 The appearing matrices $E_1, \ldots, E_{2^{j-1}}$ are of orders $u_1, \ldots, u_{2^{j-1}}$, respectively.

3.3. *Construction of the Algorithm*

In order to make our idea easily understandable by the reader and to offer some insight on how the algorithm can be developed from a software implementation point of view, we enumerate the steps carried out by the algorithm for the calculation of the $n-3$ minor of a Hadamard matrix H.

i) Write the matrix as

$$H = \begin{bmatrix} M & U_3 \\ U_3^T & D \end{bmatrix},$$

where $M,\ C$ and U_3 are described before and U_3 contains u times the column $[+,+,+]^T$, v $[+,+,-]^T$, x $[+,-,+]^T$ and y $[+,-,-]^T$. The numbers $u,\ v,\ x$ and y are found as solution of the familiar system.

ii) Calculate

$$DD^T \equiv \begin{bmatrix} E_1 & F_1 \\ F_1^T & G_1 \end{bmatrix}.$$

where $E_1 = (k_{E_1} - \lambda_{E_1})I_u + \lambda_{E_1} J_u$, $F_1 = \begin{bmatrix} f_{1,1}J_{u,v} & f_{1,2}J_{u,x} & f_{1,3}J_{u,y} \end{bmatrix}$ and $G_1 =$

$$\begin{bmatrix} (k_{1,G_1} - \lambda_{1,G_1})I_v + \lambda_{1,G_1}J_v & g_{1,2}J_{v,x} & g_{1,3}J_{v,y} \\ g_{2,1}J_{x,v} & (k_{2,G_1} - \lambda_{2,G_1})I_x + \lambda_{2,G_1}J_x & g_{2,3}J_{x,y} \\ g_{3,1}J_{y,v} & g3,2J_{y,x} & (k_{3,G_1} - \lambda_{3,G_1})I_y + \lambda_{3,G_1}J_y \end{bmatrix}$$

iii) $det(E_1)$ can be calculated from (2) and E_1^{-1} obtained from (3) can be denoted as $E_1^{-1} = (k_{E_1^{-1}} - \lambda_{E_1^{-1}})I_u + \lambda_{E_1^{-1}} J_u$.

iv) $$F_1^T E_1^{-1} = \begin{bmatrix} (f_{1,1}k_{E_1^{-1}} + (u-1)f_{1,1}\lambda_{E_1^{-1}})J_{v,u} \\ (f_{1,2}k_{E_1^{-1}} + (u-1)f_{1,2}\lambda_{E_1^{-1}})J_{x,u} \\ (f_{1,3}k_{E_1^{-1}} + (u-1)f_{1,3}\lambda_{E_1^{-1}})J_{y,u} \end{bmatrix} \equiv \begin{bmatrix} (f_1^T e_1^{-1})_1 J_{v,u} \\ (f_1^T e_1^{-1})_2 J_{x,u} \\ (f_1^T e_1^{-1})_3 J_{y,u} \end{bmatrix}.$$

v) $$F_1^T E_1^{-1} F_1 = u \begin{bmatrix} (f_1^T e_1^{-1})_1 f_{1,1} J_{v,v} & (f_1^T e_1^{-1})_1 f_{1,2} J_{v,x} & (f_1^T e_1^{-1})_1 f_{1,3} J_{v,y} \\ (f_1^T e_1^{-1})_2 f_{1,1} J_{x,v} & (f_1^T e_1^{-1})_2 f_{1,2} J_{x,x} & (f_1^T e_1^{-1})_2 f_{1,3} J_{x,y} \\ (f_1^T e_1^{-1})_3 f_{1,1} J_{y,v} & (f_1^T e_1^{-1})_3 f_{1,2} J_{y,x} & (f_1^T e_1^{-1})_3 f_{1,3} J_{y,y} \end{bmatrix}$$

$$\equiv \left[(f_1^T e_1^{-1} f_1)_{i,j} \right].$$

vi) $G_1 - F_1^T E_1^{-1} F_1 = u\cdot$

$$\begin{bmatrix} (k_{E_2}-\lambda_{E_2})I_v+\lambda_{E_2}J_v & f_{2,1}J_{v,x} & f_{2,2}J_{v,y} \\ f_{2,1}J_{x,v} & (k_{1,G_2}-\lambda_{1,G_2})I_x+\lambda_{1,G_2}J_x & g_{1,2}J_{x,y} \\ f_{2,2}J_{y,v} & g_{1,2}J_{y,x} & (k_{2,G_2}-\lambda_{2,G_2})I_y+\lambda_{2,G_2}J_y \end{bmatrix}$$

$\equiv \begin{bmatrix} E_2 & F_2 \\ F_2^T & G_2 \end{bmatrix}$, where

$k_{E_2} = k_{1,G_1} - (f_1^T e_1^{-1} f_1)_{1,1}$,
$\lambda_{E_2} = \lambda_{1,G_1} - (f_1^T e_1^{-1} f_1)_{1,1}$ $f_{2,i} = g_{1,i+1} - (f_1^T e_1^{-1} f_1)_{1,i+1}$, $i = 1, 2$,
$g_{1,2} = g_{2,3} - (f_1^T e_1^{-1} f_1)_{2,3}$,
$k_{i,G_2} = k_{i,G_1} - (f_1^T e_1^{-1} f_1)_{i,i}$, $i = 2, 3$
$\lambda_{i,G_2} = \lambda_{i,G_1} - (f_1^T e_1^{-1} f_1)_{i,i}$, $i = 2, 3$.

vii) We repeat Steps iii)-vi) similarly as above for the matrices E_2, F_2 and G_2 and obtain $G_2 - F_2^T E_2^{-1} F_2 \equiv \begin{bmatrix} E_3 & F_3 \\ F_3^T & G_3 \end{bmatrix}$, where the matrices E_3, F_3 and G_3 are appropriately specified, like in Steps 2 and 6 before.

viii) We repeat Steps iii)-vi) similarly as above for the matrices E_3, F_3 and G_3 and obtain $G_3 - F_3^T E_3^{-1} F_3 \equiv E_4 = (k_{E_4} - \lambda_{E_4}) I_y + \lambda_{E_4} J_y$.

ix) According to (4), $det\, DD^T = det\, E_1 \cdot det\, E_2 \cdot det\, E_3 \cdot det\, E_4$.

In order to implement the algorithm on the computer for this specific example, we introduce the following three matrices for the storage of all necessary variables.

$$E := \begin{bmatrix} k_{E_1} & \lambda_{E_1} & k_{E_1^{-1}} & \lambda_{E_1^{-1}} \\ k_{E_2} & \lambda_{E_2} & k_{E_2^{-1}} & \lambda_{E_2^{-1}} \\ k_{E_3} & \lambda_{E_3} & k_{E_3^{-1}} & \lambda_{E_3^{-1}} \\ k_{E_4} & \lambda_{E_4} & k_{E_4^{-1}} & \lambda_{E_4^{-1}} \end{bmatrix}, \quad res2 := \begin{bmatrix} k_{1,G_1} & k_{2,G_1} & k_{3,G_1} \\ 0 & k_{1,G_2} & k_{2,G_2} \\ 0 & 0 & k_{1,G_3} \end{bmatrix} \text{ and}$$

$$res3 := \begin{bmatrix} \lambda_{1,G_1} & \lambda_{2,G_1} & \lambda_{3,G_1} \\ 0 & \lambda_{1,G_2} & \lambda_{2,G_2} \\ 0 & 0 & \lambda_{1,G_3} \end{bmatrix}.$$

A matrix denoted by $res1$ stands for DD^T, i.e. $DD^T = (n - j - a_{ii}) I_{u_1,u_2,\cdots,u_{2j-1}} + a_{ik} J_{u_1,u_2,\cdots,u_{2j-1}} \equiv res1$.

3.4. *A note on complexity*

The Algorithm is designed in such a way, that the special structure and properties of every appearing matrix are taken into account. So, all necessary matrix multiplications and inversions are not performed explicitly but in an efficient manner so that the total computational cost remains at low levels. A closer examination of the proposed algorithm and its construction reveals that the computational cost is

$t_1 = 2^{j-1} \cdot O(2^{2(j-1)}) = O(2^{3(j-1)})$(the outer factor 2^{j-1} corresponds to the for-loop in Step 5) operations for matrix multiplications and inversions, carried out in Steps 4-6, where the core of the computational work of the algorithm takes place. This amount is significantly less than the number of flops required, if the calculations are performed explicitly, even if the fastest possible algorithm for matrix multiplication and inversion is used (see [7],[6] and the references therein). A fast matrix multiplication method forms the product of two $n \times n$ matrices in $O(n^\omega)$ arithmetic operations, where $\omega < 3$. Coppersmith and Winograd [1] introduced an algorithm for this purpose with $\omega = 2.376$, which is the best value known today. Since the product C of two $n \times n$ matrices A and B consists of n^2 elements (i.e. each element of each matrix must partake in at least one operation), it is obvious that there cannot exist an algorithm for the calculation of C with complexity better than $O(n^2)$, i.e. $\omega \geq 2$. So, we can say in other words that our algorithm attains the optimal value ω for the complexity of the appearing matrix multiplications. Of course, this algorithm deals with very special structured matrices. Until now it is unknown, whether such an algorithm with this optimal complexity can exist for an arbitrary general matrix.

Taking these facts into account, we can conclude that the best computational cost, which can be achieved for our algorithm using the ordinary matrix product with the optimal ω, is $t_2 = \frac{z^4}{3} + z^{3.376}$, where $z = 2^{j-1}$. Precisely, the cost for Steps 4-6 consists of $O(\frac{(2^{j-1})^3}{3})$ flops for each inversion, $O(2^{(j-1)\cdot 2.376})$ flops for each matrix multiplication and due to the for-loop in Step 5, we have totally $2^{j-1}(\frac{(2^{j-1})^3}{3} + 2^{(j-1)\cdot 2.376})$ flops. Obviously, $t_2 > t_1$, as it can be also seen in Figures 1 and 2, where t_1, t_2 are plotted as functions of j for various values of j. For $j < 20$ there is no significant difference between t_1 and t_2, but t_1 is always less than t_2.

Although the algorithm might seem complicated, it contains only the absolutely necessary computations for our purpose, which were firstly carried out by hand taking always into account (2), (3) and (4), and in the sequel the results of these computations were embedded inside the algorithm. Regardless of all this effort and although the minimum possible cost for matrix multiplication is reached, the total computational cost of the algorithm remains at high levels, because of the exhaustive (complete) searches performed at Steps 2 and 3. Unfortunately these searches cannot be avoided because if some cases are excluded we might lose some values of minors, which could probably appear and in this way the result is false, since we will not have calculated all possible values.

3.5. *Implementation of the Algorithm*

We provide exemplarily the results of Algorithm Minors for its implementation with $j = 4$ and $n = 16$. We suppose initially that only the following matrix (or an H-equivalent to it) can appear in the upper left 4×4 corner of a CP Hadamard matrix,

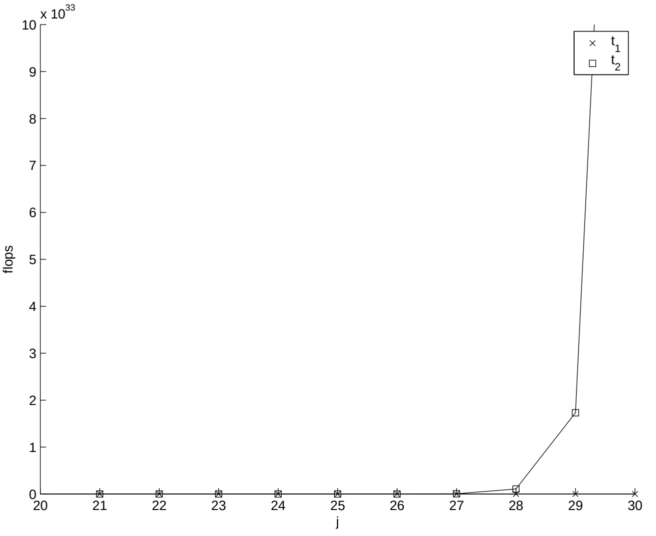

Figure 1: Computational costs t_1 and t_2 vs. j.

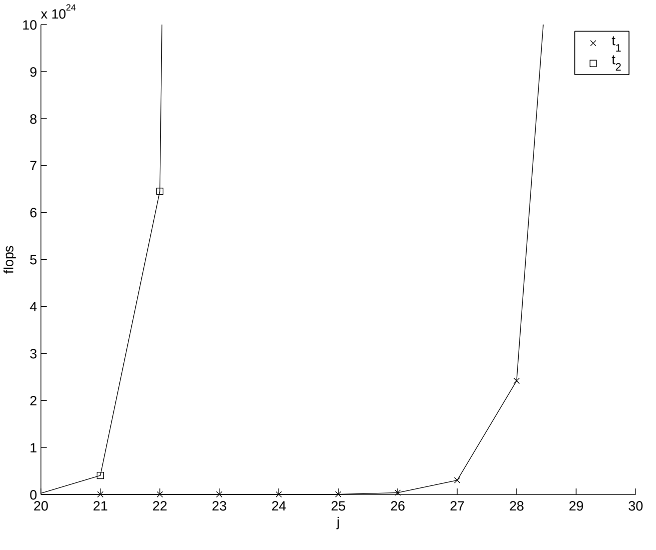

Figure 2: Computational costs t_1 and t_2 vs. j, zoomed picture.

as it was already stated in [10]:

$$M = \begin{bmatrix} + & + & + & + \\ + & - & + & - \\ + & + & - & - \\ + & - & - & + \end{bmatrix}.$$

We have

$$U_4 = \begin{array}{cccccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \\ + & + & + & + & + & + & + & + \\ + & + & + & + & - & - & - & - \\ + & + & - & - & + & + & - & - \\ + & - & + & - & + & - & + & - \end{array}$$

The system of 7 equations and 8 variables, which results from counting of columns and the orthogonality of every two distinct rows of the matrix $[M \quad U_4]$, is

$$\begin{aligned} u_1 + u_2 + u_3 + u_4 + u_5 + u_6 + u_7 + u_8 &= n - 4 \\ u_1 + u_2 + u_3 + u_4 - u_5 - u_6 - u_7 - u_8 &= 0 \\ u_1 + u_2 - u_3 - u_4 + u_5 + u_6 - u_7 - u_8 &= 0 \\ u_1 + u_2 - u_3 - u_4 - u_5 - u_6 + u_7 + u_8 &= 0 \\ u_1 - u_2 + u_3 - u_4 + u_5 - u_6 + u_7 - u_8 &= 0 \\ u_1 - u_2 + u_3 - u_4 - u_5 + u_6 - u_7 + u_8 &= 0 \\ u_1 - u_2 - u_3 + u_4 + u_5 - u_6 - u_7 + u_8 &= 0 \end{aligned}$$

The solution is:

$$\begin{aligned} u_1 &= 3 - u_8 \\ u_2 &= u_8 \\ u_3 &= u_8 \\ u_4 &= 3 - u_8 \\ u_5 &= u_8 \\ u_6 &= 3 - u_8 \\ u_7 &= 3 - u_8 \\ u_8 &= u_8 \end{aligned}$$

According to Lemma 3.1, the parameter u_8 can take the values $0, 1, 2, 3, 4$. We present only the case $u_8 = 1$, since the rest of them can be handled absolutely similarly. By denoting with D the remaining matrix after deleting the first four rows and columns of the initial Hadamard matrix, we have:

$$DD^T \equiv \begin{bmatrix} E_1 & F_1 \\ F_1^T & G_1 \end{bmatrix},$$

where $E_1 = nI_{u_1} - 4J_{u_1}$,
$F_1 = [-2_{u_1\times u_2} \ -2_{u_1\times u_3} \ 0_{u_1\times u_4} \ -2_{u_1\times u_5} \ 0_{u_1\times u_6} \ 0_{u_1\times u_7} \ 2_{u_1\times u_8}]$ and $G_1 =$

$$\begin{bmatrix} E_{1u_2\times u_2} & 0 & 2 & 0 & 2 & 2 & 0 \\ 0 & E_{1u_3\times u_3} & 2 & 0 & 2 & 2 & 0 \\ 2 & 2 & E_{1u_4\times u_4} & 2 & 0 & 0 & 2 \\ 0 & 0 & 2 & E_{1u_5\times u_5} & 2 & 2 & 0 \\ 2 & 2 & 0 & 2 & E_{1u_6\times u_6} & 0 & 2 \\ 2 & 2 & 0 & 2 & 0 & E_{1u_7\times u_7} & 2 \\ 0 & 0 & 2 & 0 & 2 & 2 & E_{1u_8\times u_8} \end{bmatrix}.$$

For the sake of better presentation we introduce 2 standing for -2 and we omit the subscripts of the elements $0, \pm 2$ in G_1, which actually represent blocks with these elements of appropriate size.

We have $det\, E_1 = (n-8) \cdot n$ and the above explained calculations give $G_1 - F_1^T E_1^{-1} F_1 \equiv \begin{bmatrix} E_2 & F_2 \\ F_2^T & G_2 \end{bmatrix}$. From now on, all the appearing intermediate matrices are obtained according to the already described method and it is sensible not to give all of them analytically. For all the remaining steps, with help of equation (2), we obtain

$det\, E_2 = \frac{n^2 - 12n + 24}{n-8}$, $det\, E_3 = \frac{(n-4)(n^2-10n+16)}{n^2-12n+24}$,

$det\, E_4 = \frac{(n-8)(n-12)n^2}{n^2-12n+16}$, $det\, E_5 = \frac{n^2-12n+16}{n-8}$,

$det\, E_6 = \frac{(n-12)(n^2-12n+24)n^2}{(n-4)(n^2-12n+16)}$, $det\, E_7 = \frac{(n-8)(n-12)n^2}{n^2-12n+24}$

and $det\, E_8 = \frac{(n-12)n}{n-8}$.

Finally, $det\, DD^T = \prod_{i=1}^{8} det\, E_i = n^8 \cdot (n-12)^4 = 1099511627776.$

So $det\, D = \sqrt{det\, DD^T} = 1048576$ is the only possible appearing value for the minors of order $n-4$ for a CP Hadamard matrix of order 16. We mention that this resulting value is subject to the formula $16n^{\frac{n}{2}-4}$.

A consequence of the idea of the construction of Algorithm Minors is that it can be either of symbolical or numerical nature, according to the purpose of the user and after the appropriate settings are made in the computer programm. If the user's intention is to obtain general formulas containing n, then there must *not* be assigned to n a fixed value after Step 4 (included), while in Step 3 it is mandatory that n has a specified value because there exists the expression $\frac{n}{4}$ as end value of the for-loop. In this case the algorithm's function is symbolical and the discussion done in §3.4 does not apply, since symbolical computations are associated only with time of execution and not with floating point operations. For example, in the above presented implementation, the value $n = 16$ was used initially to provide the end value $\frac{n}{4} = 4$ of the possible parameter's values. Afterwards, n is considered general, all the intermediate results are obtained symbolically and the value 16 is assigned again to n only at the end, in order to calculate the exact requested value of the minor.

If the user's intention is to calculate the value of the minor for a specific n fixed, then n must be given at the beginning of the programm, so it attains a specific value throughout the whole programm, and hence all calculations are done in floating point arithmetic, i.e. in a numerical sense, and they are subject to the conclusions of §3.4. The main advantages of each case are that the symbolic implementation is absolutely accurate and doesn't contain any errors due to possible roundoffs and that the numeric implementation is significantly faster.

4. Experimental Results

For the needs of this study, we tested Algorithm Minors for $n = 12,\ 16$ and for $j = 1, \ldots, 7$. The results obtained from the numerical experiments performed with algorithm Minors are subject to the formulas presented in Table 1.

order	values of minors
$n-1$	$n^{n/2-1}$
$n-2$	0, $2n^{n/2-2}$
$n-3$	0, $4n^{n/2-3}$
$n-4$	0, $8n^{n/2-4}$, $16n^{n/2-4}$
$n-5$	0, $16n^{n/2-5}$, $32n^{n/2-5}$, $48n^{n/2-5}$
$n-6$	0, $32n^{n/2-6}$, $64n^{n/2-6}$, $96n^{n/2-6}$, $128n^{n/2-6}$, $160n^{n/2-6}$
$n-7$	0, $64n^{n/2-7}$, $128n^{n/2-7}$, $192n^{n/2-7}$, $256n^{n/2-7}$, $320n^{n/2-7}$
	0, $384n^{n/2-7}$, $448n^{n/2-7}$, $512n^{n/2-7}$, $576n^{n/2-7}$

Table 1. Values of minors of orders $n-1, \ldots, n-7$ for Hadamard matrices of order $n = 12, 16$.

We observe that all possible values of the $(n-j) \times (n-j)$ minors are 0 or $p \cdot n^{(n/2)-j}$, for $p = 2^{j-1}, 2 \cdot 2^{j-1}, 3 \cdot 2^{j-1}, \ldots, s \cdot 2^{j-1}$, where $s \cdot 2^{j-1} = max\{det(A) | A \in \mathbb{R}^{j\times j}, \text{with elements} \pm 1\}$ and the value 0 is excluded from the $n-1$ case.

Alternatively, the obtained results can be summarized according to the formula

$$M_{n-j} = 0 \text{ or } p \cdot n^{(n/2)-j},\ j = 0, 1, 2, \ldots,$$

where for the evaluation of the coefficient p the following procedure is adopted:

$p := 2^{j-1}$
$m := max\{det(A) | A \in \mathbb{R}^{j\times j}, \text{with elements} \pm 1\}$
$k := 1$
repeat
  $p = k \cdot p$
  $k = k + 1$
until
  $p = m$.

The maximum determinant values for ± 1 $n \times n$ matrices are given in the following Table.

n	1	2	3	4	5	6	7
max. det.	1	2	4	16	48	160	576

Table 2. Maximum determinant values for $n \times n$ ± 1 matrices.

5. Conclusion

We presented the theoretical background of the required determinant manipulations for calculating the minors of order $n-j$ of a Hadamard matrix, which finally leads to the construction of algorithm Minors. The algorithm, which has interesting complexity properties, is designed by taking considerably into account the structure of Hadamard matrices and can be applied theoretically for every value of n and j.

A more sophisticated version of the proposed algorithm is currently under investigation, which could eventually work in a sense of parallel implementation. The usefulness of such an algorithm lies in the fact, that it can be utilized after appropriate modifications in combination with relation (1) for the calculation of the pivot structure of the Hadamard matrix of order 16, which still remains an unsolved problem.

References

1. D. Coppersmith and S. Winograd, Matrix Multiplication via arithmetic progressions, *J. Symbolic Comput.*, **9** (1990), pp. 251–280.
2. C.W. Cryer, Pivot size in Gaussian elimination, *Numer. Math.*, **12** (1968), pp. 335–345.
3. J. Day and B. Peterson, Growth in Gaussian Elimination, *Amer. Math. Monthly*, **95** (1988), pp. 489–513.
4. I. Z. Emiris and V. Y. Pan, Improved algorithms for computing determinants and resultants, *Journal of Complexity*, **21** (2005), pp. 43–71.
5. R. Flowe and G. Harris, A Note on Generalized Vandermonde Matrices, *SIAM J. Matrix Anal. Appl.*,**14**, (1993), pp. 1146–1151.
6. G. Hämmerlin and K.-H. Hoffmann, *Numerische Mathematik*, Springer, Berlin-Heidelberg (1994).
7. N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia (2002).
8. C. Kravvaritis, M. Mitrouli and J. Seberry, Counting Techniques Specifying the Existence of Submatrices in Weighing Matrices, *Lecture Notes in Comput. Sci.*, **3718** (2005), pp. 294–305.
9. C. Krattenthaler, Advanced determinant calculus: A complement, *Linear Algebra Appl.*,**411** (2005), pp. 68–166.
10. J. Seberry, T. Xia, C. Koukouvinos and M. Mitrouli, The maximal determinant and subdeterminants of ± 1 matrices, *Linear Algebra Appl.*, **373** (2003), pp. 297–310.

◇ Christos Kravvaritis
Department of Mathematics
University of Athens
15784 Panepistimiopolis, Athens, GREECE
`ckrav@math.uoa.gr`

◇ Marilena Mitrouli
Department of Mthematics
University of Athens
15783 Panepistimiopolis, Athens, GREECE
`mmitroul@math.uoa.gr`