Some algorithms of computer algebra

O. Porkuian, A. Timoshyn, L. Timoshyna

Volodymyr Dahl East-Ukrainian National University, e-mail: timxxvii@gmail.com

Summary. The article describes one of the methods for computing determinants without using fractions proposed by Bareiss. This problem has a clear algorithmic character in nature and refers to the field of computer algebra. The implementation of this algorithm is proposed in the known Maxima system of symbolic computations. In addition, this method makes it possible to get enough convenient formula for the calculation of the matrix of unitriangular transformation of a quadratic form to a canonical one.

Key words: computer algebra, Maxima system, determinant, Bareiss algorithm, a quadratic form, unitriangular transformation.

INTRODUCTION

Modern computer algebra is inextricably linked with the use of computer processing technology, and especially with the software, which includes applied mathematical packages. Such packages as MatLab and Scilab are powerful professional mathematical packages. The essential difference between these packages is that Scilab is open source software, while MatLab being a commercial product [1, 2]. These packages have much in common, particularly, they inherently have a tendency for performing numerical calculations. Although MatLab includes many tools of symbolic computations, yet the most effective up-today systems of analytical computation are Maple, Maxima, Mathematica. For example, Maxima system has a modern user interface, powerful visualization tools of all phases of operation, a wide range of functions and special packages. The packages for matrix computations are especially useful [3-5].

However, built-in computer algebra system functions are not sufficient in some cases. There is a set of problems, such as calculating a determinant of the matrix without using fractions, the solutions of which are connected with certain algorithms, i.e. a set of commands and functions. The extensive use of matrix algebra in solving economic and technical problems makes this task even more urgent [6, 7].

THE ANALYSIS OF RECENT RESEARCHES AND PUBLICATIONS

Not so much literature is devoted to algorithms for computer algebra associated with the matrix analysis. Most of the sources [8-12] on computer algebra contain materials relating to the issues of numbers representation, polynomials, rational and algebraic functions, polynomial simplification of formal integration. Various operations with such objects assume symbolic computation. In some literature computer algebra is referred as a branch of mathematics lying on the intersection of algebra and numerical methods. Indeed, there are many problems of algebra and mathematical analysis, which are connected with symbolic computation [13-17]. The direct analysis of these issues related to matrices, and, in particular, the special algorithms for computing determinants, is presented in [18].

OBJECTIVES

The main aims of this study are:

- to study an algorithm for computing the determinant of a matrix without using fractions proposed by Bareiss, followed by the implementation of this algorithm in Maxima system.

- to investigate, on the basis of this algorithm, the reduction of a quadratic form to a canonical one by means of unitriangular transformation, or rather, to get convenient formula for calculating the unitriangular transformation matrix.

THE MAIN RESULTS OF THE RESEARCH

Let us consider the so-called *dense* matrices. A clear definition of dense matrices can be given to the aspect of *sparse* matrices. The matrix of order n is called sparse, if the number of its non-zero elements does not exceed n^{1+q} where $q < 1$. So for the sparse matrix of order 50 (with $q = 0.5$) the number of its non-zero elements equals to about 350, which accounts to a small percentage of the total number of matrix elements. As a rule, when dealing with sparse matrices, computer algebra uses representations, in which each row of the matrix is defined by a list of non-zero elements of the row, each being stored in memory, with indicating the number of its column. Now it is possible to say that the dense matrices are those that do not belong to sparse ones.

In the systems of computer algebra dense matrices are defined by rows. In particular, in Maxima system the matrix function (the call syntax: *matrix(row1, ..., rown)*) is used for presetting matrices. The presence of symbolic elements in these matrices can lead to serious problems of "swelling" the data, both the intermediate and the final ones.

Another serious problem is connected with the division that occurs, for example, when calculating

determinants. The idea is that the calculation of the determinant of the matrix as the sum of the products of the matrix elements, taken one by one from each row and each column (the method of calculating the determinant is sometimes called Cramer's rule), numerically very inefficient: the number of operations in this case is $O(n(n!))$, while the calculation of determinants by Gaussian elimination algorithm the number of operations is $O(n^3)$. It is obvious, that Gaussian elimination algorithm requires division which can lead to fractions. In addition, for matrices with elements of a ring with *zero divisors* the successive elimination method may be simply impossible. This may occur when the elements of the matrix has no unit divisors. For example, in the residue-class ring by modulo 6, the determinant of the matrix $\begin{pmatrix} 4 & 3 \\ 3 & 4 \end{pmatrix}$ equals to 1, however, elements 3 and 4 in ring Z_6 have no unit divisors, making it impossible to divide, i.e. to apply the algorithm of elimination.

As noted in [18], there is a whole family of elimination methods without using fractions, i.e., those where all appropriate division are performed accurately. Let us consider in detail a step-by-step algorithm for computing determinants without using fractions, proposed by Bareiss, which is based on generalization of Sylvester's identity [19].

We introduce some notation. Let a square matrix be given $A = (a_{ij})$ of order n, where the matrix elements being integers. Consider the determinant of the following form:

$$\Delta(k,i,j) = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1k} & a_{1j} \\ a_{21} & a_{22} & \dots & a_{2k} & a_{2j} \\ \dots & \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kk} & a_{kj} \\ a_{i1} & a_{i2} & \dots & a_{ik} & a_{ij} \end{vmatrix}. \quad (1)$$

Note that the determinant $\Delta(k,i,j)$ is obtained by bordering the *i*-th row and the *j*-th column of the upper-left corner (main) minor Δ_k of order k of matrix A, where $n \geq i, j > k$.

The basic ratio of the step-by-step algorithm for computing determinants without using fractions has the following form:

$$\Delta(k,i,j) = \frac{1}{\Delta(k-2,k-1,k-1)} \times$$
$$\times \begin{vmatrix} \Delta(k-1,k,k) & \Delta(k-1,k,j) \\ \Delta(k-1,i,k) & \Delta(k-1,i,j) \end{vmatrix}. \quad (2)$$

It is obvious, that the value of the determinant $\Delta(k,i,j)$ is an integer (from (1) it can be computed by Cramer's rule). Hence, the right side of (2) is an integer, i.e. the division is performed without a remainder.

This makes it possible to calculate the determinant of the matrix *A* in step-by-step way, the intermediate results

being integers. According to (1), $\det(A) = \Delta(n-1,n,n)$. On the other hand, according to (2):

$$\det(A) = \Delta(n-1,n,n) =$$
$$= \frac{1}{\Delta(n-3,n-2,n-2)} \times$$
$$\times \begin{vmatrix} \Delta(n-2,n-1,n-1) & \Delta(n-2,n-1,n) \\ \Delta(n-2,n,n-1) & \Delta(n-2,n,n) \end{vmatrix}.$$

We are dealing with a recursive algorithm, which in Maxima system can be implemented as follows:

```
determ[n,i,j]:=
if n=1 then A[n,n]*A[i, j]–A[i,n]*A[n,j] else
if n=2 then (1/A[n–1,n–1])*(determ[n–1,n,n]*
determ[n–1,i,j]–determ[n–1,n,j]*determ[n–1,i,n]) else
(1/determ[n–2,n–1,n–1])*(determ[n–1,n,n]*
determ[n–1,i,j]–determ[n–1,n,j]*determ[n–1,i,n])$.
```

This algorithm is a relative of the algorithms of elimination elements. Indeed, consider an arbitrary symbolic square matrix:

$$A = ([a11, a12, \dots], [a21, a22, \dots], [a31, a32, \dots], \dots).$$

Let us perform the following elementary transformations of the rows of this matrix (as a rule, such a transformation is performed at a reduction of a matrix to echelon form for calculating the rank of a matrix or solving the system of linear equations). We multiply the first row of the matrix to the element $a21$, and the second row to the element $(-a11)$, and then add the first row to the second one. The second row will look:

$$[0, \ a11a22 - a12a21, \ a11a23 - a13a21, \dots].$$

The resulting second row in the notation of (1) can be rewritten as follows:

$$[0, \ \Delta(1,2,2), \ \Delta(1,2,3), \dots].$$

Similarly, we set to zero the element $a31$ and repeat the procedure for the newly obtained second and third rows. We have

$$[0, \ 0, \ a11 \cdot \Delta(2,3,3), \dots].$$

It is obvious, that all of the remaining elements of the third row are divided by $a11$. A similar situation occurs for the following rows, which provides integer results.

Let us consider one more algebraic problem, which is reduced to a recursive algorithm, while again there occur the determinants of type $\Delta(k,i,j)$.

Let $f(X) = \sum_{i,j=1}^{n} a_{ij} x_i x_j$ be a quadratic form, where $A = (a_{ij})$ being a matrix of a quadratic form. Lagrange's method, the eigenvectors method and Jacobi's method can be referred as the most well-known methods of reducing a quadratic form to a canonical one. The reduction problem

involves finding the canonical coefficients and the corresponding transformation, i.e. a matrix B that $A' = B^T A B$ being a diagonal matrix. For Lagrange's method and the method of eigenvectors the reduction problem is fairly well presented in any textbook of algebra. However, when using the Jacobi's method (which is also called the method of unitriangular transformation), as a rule, researchers confine themselves to finding the canonical coefficients. Constructing an appropriate transformation is based on the following theorems [20].

Theorem 1. Let a matrix of a quadratic form be non-degenerate. In order to transform it to a canonical form using a right unitriangular matrix, it is necessary and sufficient that the principal minors of the matrix of the quadratic form be non-zero.

Theorem 2. For the presentation of a non-degenerate matrix $A = (a_{ij})$ of order n as a product of the left unitriangular, diagonal and right unitriangular matrix, i.e. $A = LDR$, it is necessary and sufficient that the principal minors of the matrix be non-zero.

Since the matrix of a quadratic form is symmetric, $A = R^T D R$. Introduce the following notation:

$$A_k = \begin{pmatrix} A_{k-1} & a_k \\ b_k & a_{kk} \end{pmatrix}, \quad R_k = \begin{pmatrix} R_{k-1} & X_k \\ 0 & 1 \end{pmatrix},$$

$$D_k = \begin{pmatrix} D_{k-1} & 0 \\ 0 & d_k \end{pmatrix},$$

where: $n \ge k \ge 2, \quad b_k = a_k^T$.

Provided that $A = R^T D R$, the equality $A_k = R_k^T D_k R_k$ is true. This leads to the following relationships:

$$\begin{cases} d_k = \dfrac{\det A_k}{\det A_{k-1}} = \dfrac{\Delta_k}{\Delta_{k-1}}, \\ X_k = D_{k-1}^{-1}(R_{k-1}^T)^{-1} a_k \end{cases}$$

where: $D_1 = d_1 = a_{11}, R_1 = 1$, X_k is an unknown column.

In case of $n = 3$, we have:

$$X_2 = d_1^{-1} \cdot a_{12} = \frac{a_{12}}{\Delta_1}. \quad R_2 = \begin{pmatrix} 1 & \dfrac{a_{12}}{\Delta_1} \\ 0 & 1 \end{pmatrix},$$

$$X_3 = D_2^{-1} \cdot (R_2^T)^{-1} \cdot [a_{13} a_{23}]^T = \begin{pmatrix} \dfrac{a_{13}}{\Delta_1} \\ \dfrac{a_{11}a_{23} - a_{12}a_{13}}{d_1 d_2} \end{pmatrix},$$

$$R_3 = \begin{pmatrix} 1 & \dfrac{a_{12}}{\Delta_1} & \dfrac{a_{13}}{\Delta_1} \\ 0 & 1 & \dfrac{\Delta(1,2,3)}{\Delta(1,2,2)} \\ 0 & 0 & 1 \end{pmatrix}.$$

Thus, we get enough convenient formula for calculating the matrix (in fact, it is a matrix R^{-1}) of unitriangular transformation of a quadratic forms to a canonical one for dimension 3.

Further calculations show that when increasing the dimension the matrix R has the following form:

$$R = \begin{pmatrix} 1 & \dfrac{a_{12}}{\Delta_1} & \dfrac{a_{13}}{\Delta_1} & \cdots & \dfrac{a_{1j}}{\Delta_1} & \cdots \\ 0 & 1 & \dfrac{\Delta(1,2,3)}{\Delta(1,2,2)} & \cdots & \dfrac{\Delta(1,2,j)}{\Delta(1,2,2)} & \cdots \\ 0 & 0 & 1 & \cdots & \dfrac{\Delta(2,3,j)}{\Delta(2,3,3)} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix}.$$

CONCLUSIONS

The following important conclusions can be drawn from this study:

1. As a result of analyzing the known packages of computer algebra the decision of using Maxima system in further investigations has been made. This system has all functions and libraries necessary for performing matrix computations.

2. Learning different literature on computer algebra enables to state that insufficient attention is paid to special algorithms of matrix computations. The problems of implementing these algorithms in the systems of computer algebra are not being practically discussed.

3. Based on the step-by-step algorithm for computing determinants without using fractions, proposed by Bareiss, the recursive procedure for computing minors of the type $\Delta(k,i,j)$ in Maxima system has been constructed.

4. More detailed study of Jacobi's method (the method of transformation of a quadratic form to a canonical one) also leads to minors of the type $\Delta(k,i,j)$. In particular, the convenient formula for computing matrices of unitriangular transformation, which uses minors of the type $\Delta(k,i,j)$ has been obtained in this study.

REFERENCES

1. Some systems symbolic computation open-source. Available online at: < http://cph.phys.spbu.ru/>
2. **Klimenko V. P., Lyakhov A. L., Gvozdik D. N. 2011.** Modern features of computer algebra systems. Journal Mathematical Machines and Systems. Vol. 1. Nr 2, 3-18. (in Russia).

3. **Chichkarev E. A. 2009.** Computer mathematics with Maxima: Guide pupils and students. – M.: ALT Linux, 2009. (in Russia).

4. **Stahin N. A. 2008.** Basics of system analysis (symbolic) Maxima computing (software for solving analytical problems computation). Tutorial. – M.: Mir, – 86. (in Russia).

5. Maxima system of computer algebra. Available online at: < http://maxima.sourceforge.net>.

6. **Sergiy Gutyrya, Victor Yaglinsky. 2006.** System criteria an alysis and function optimization of industrial robots // TEKA. Commission of Motorization and Power Industry in Agriculture, Poland. – Lublin: Polish Academy of Sciences Branch in Lublin, 2006. – V. 6A, 70-81. (in English).

7. **Valentin Shishov, Pavel Nosko, Pavel Fil, Alexandr Muhovatiy, Julia Sklyar. 2008.** Parametrical optimization of worm gears based on losses in gearing // TEKA. Commission of Motorization and Power Industry in Agriculture, Poland. – Lublin: Polish Academy of Sciences Branch in Lublin, 2008. – V. 8, 213-221. (in English).

8. **Kuznetsov M. I., Burlankov D. E., Dolgov G. A., Chirkov A., Yakovlev V. A. 2002.** Computer Algebra. – N. Novgorod. Publishing House of the Nizhny Novgorod State University Press, 223. (in Russia).

9. **Pankratiev E. V. 2010.** Elements of computer algebra. – M .: Binom, – 248. (in Russia).

10. **Buchberger B., Collins G., Loos R.. 1983.** Computer Algebra. Symbolic and Algebraic Computation. – By Springer Verlag/Wien, – 288. (in English).

11. **Cooke D. J., Bez H. E. 1984.** Computer Mathematics. – Cambridge University Press, – 408. (in English).

12. **Roman Dmytryshyn. 2011.** Accuracy of the Hilbert matrix determinant calculation. 12th International Workshop "Computational Problems of Electrical Engineering", 19-20. (in English).

13. **Takuya Kitamoto. 2014.** On the Computation of the Determinant of a Generalized Vandermonde Matrix // Vladimir P. Gerdt, Wolfram Koepf, Werner M. Seiler, Evgenii V. Vorozhtsov. Computer Algebra in Scientific Computing: 16th International Workshop, CASC 2014 Warsaw, Poland, September 8-12, 2014 Proceedings. 242-255. (in English).

14. Fermat is a computer algebra system for polynomial and matrix computation. Available online at: <http://home.bway.net/lewis/ >.

15. **Kh. D. Ikramov, M. Matin far, A. A. Chesnokov. 2004.** Computer algebra procedures for the Drazin inversion of a singular matrix. Comput. Math. Math.. Phys, 44: 7 (2004), 1155-1163. (in English).

16. **Ikramov Kh. D. , Matin M. far. 2003.** On computer algebra procedures for matrix pseudoinversion, Zh. Vychisl. Mat. Mat. Fiz., **43**:2 (2003), 163–168. (in Russia).

17. **Akritas A. 1994.** Elements of Computer Algebra with Applications / Trans. from English. - M.: Mir, - 544. (in Russia).

18. **Davenport J., Siret Y., Tournier E.. 1991.** Calcul formel: Symbolic and algebraic computation / Trans. from English. – M.: Mir, – 350. (in Russia).

19. **Prasolov V. V. 2008.** Problems and theorems of linear algebra. – M.: Nauka, – 269. (in Russia).

20. **Faddeev D. K. 1984.** Lectures on algebra. Textbook for high schools. – M .: Nauka, – 416. (in Russia).

НЕКОТОРЫЕ АЛГОРИТМЫ КОМПЬЮТЕРНОЙ АЛГЕБРЫ

О.В. Поркуян, А.С.Тимошин, Л.В. Тимошина

Аннотация. В статье рассмотрен один из методов вычисления определителей без использования дробей, предложенный Барейсом. Эта задача имеет четко выраженный алгоритмический характер и относится к разделу компьютерной алгебры. Реализация соответствующего алгоритма предлагается в известной системе символьных вычислений Maxima. Кроме того, этот метод дает возможность получить достаточно удобную формулу для расчета матрицы унитреугольного преобразования квадратичной формы к каноническому виду.

Ключевые слова: компьютерная алгебра, система Maxima, определитель, алгоритм Барейса, квадратичная форма, унитреугольное преобразование.