ISSN 0361-7688, Programming and Computer Software, 2021, Vol. 47, No. 1, pp. 50–55. © Pleiades Publishing, Ltd., 2021.
Russian Text © The Author(s), 2021, published in Programmirovanie, 2021, Vol. 47, No. 1.

Heuristic Algorithms for Recognition of Some Cubic Hypersurfaces

A. V. Seliverstov

Institute for Information Transmission Problems (Kharkevich Institute), Russian Academy of Sciences, Bol'shoi Karetnyi per. 19, Moscow, 127051 Russia

e-mail: slvstv@iitp.ru

Received June 29, 2020; revised August 26, 2020; accepted September 12, 2020

Abstract—In this paper, we propose some heuristic probabilistic polynomial time algorithms with one-sided error for recognition of cubic hypersurfaces the singular loci of which do not contain any linear subspace of sufficiently large dimension. These algorithms are easy to implement in computer algebra systems. The algorithms are based on checking the condition that the Hessian determinant of a cubic form does not vanish identically or does not determine any cone in the projective space. In turn, the properties of the Hessian can be verified with one-sided-error probabilistic algorithms based on the Schwartz–Zippel lemma.

DOI: 10.1134/S0361768821010096

1. INTRODUCTION

A hypersurface is a projective variety defined by one form (homogeneous polynomial). A curve on a plane and a surface in a 3D space are hypersurfaces. It is assumed that the hypersurface is defined by a square-free form over a field of characteristic zero in which arithmetic operations are computable in a polynomial number of bit operations, e.g., over the field of rational numbers [1]. A point is called singular if all first partial derivatives of this form vanish. Finding a singular point reduces to finding a nontrivial solution to a system of homogeneous algebraic equations. In small dimensions, this problem is easy to solve by symbolic computations, e.g., in Maple or MathPartner [2]. However, in high dimensions, the problems of smoothness recognition and finding a singular point on a cubic hypersurface are algorithmically complex.

When discussing computational complexity, we consider symbolic computations with no rounding of numerical values and no rational approximation of meromorphic functions. When expanding the field of rational numbers, computer algebra methods cannot be replaced by numerical methods. Unless otherwise stated, the runtime of an algorithm is determined by the number of arithmetic operations in a field, the number of comparison and copying operations, as well as the number of operations on indices. More formally, we consider computations on generalized register machines over a certain field [3]. When each arithmetic operation on a field is computable in a polynomial number of bit operations, the polynomial time computability on a generalized register machine over this field, generally speaking, does not imply the computability in a polynomial number of bit operations because the solution can have exponential length. For instance, this effect occurs with multiple exponentiation $(\dots(x^2)^2\dots)^2$.

In the worst case, the problem is complex; in the typical case, it can be easier to solve [4]. In this connection, the so-called generic algorithms are of interest: these algorithms quickly find a correct solution for almost all inputs (for a large number of inputs of a given length); however, for a small number of inputs, they only issue a warning about infeasibility of computations [5, 6]. A generic algorithm always yields correct solutions in contrast to a heuristic algorithm, which yields correct solutions for almost all inputs, however, for some inputs, it can yield false solutions. An algorithm with a one-sided error can make errors of only one kind. A trivial algorithm that accepts all inputs is an example of a heuristic algorithm with one-sided error for the problem of recognizing the smoothness of a cubic hypersurface because a smooth hypersurface is defined by a general form. For this problem, it is difficult to propose a heuristic algorithm with one-sided error of another kind that recognizes singular cubic hypersurfaces without errors.

Probabilistic algorithms use random bits [4]. An example is the probabilistic reduction of finding a binary solution to a system of algebraic equations to finding a binary solution of one equation [7]. Replacing random bits by a pseudo-random sequence turns a probabilistic algorithm to a heuristic one (but, generally speaking, not to a generic one).

There are several methods for finding singular points. The operation of finding a solution to a system of algebraic equations by computing the Gröbner basis is implemented in many computer algebra systems; however, it requires a lot of time and memory [8, 9]. Investigating the properties of a singular point involves the Newton polytope [10]. This polytope was first used by A.D. Bryuno to find asymptotics of solutions to systems of differential equations [11]; he also used its generalization—the Hadamard polytope—for parameterization [12, 13]. For a general hypersurface with only one singular point, which acts as an ordinary double point, the coordinates of this singular point are expressed in terms of rational functions of the coefficients of a form that defines this hypersurface [14].

A hypersurface of degree d in $\mathbb{P}^n$ corresponds to a hyperplane section of a Veronese variety. A hypersurface is singular if the secant hyperplane touches the Veronese variety, i.e., acts as a point of a dual variety. The degree of this dual variety is $(n+1)(d-1)^n$. It is the degree of the discriminant of a d-degree form in $n + 1$ variables [15]. For instance, the discriminant of a quadratic form is proportional to the determinant of a Hessian matrix, and its degree is equal to the number of variables $n + 1$. If a binary form acts as a homogenization of an inhomogeneous polynomial in one variable, then the discriminant of the form vanishes when this polynomial has a multiple root.

Smoothness of plane curves can be checked using special methods [16]. A cubic curve is projectively equivalent to a curve in the Weierstrass form. Its affine part is given by the equation $y^2 = x^3 + px + q$. This curve is singular when the discriminant of the polynomial on its right-hand side, which is equal to $-4p^3 - 27q^2$, vanishes. Reduction to the Weierstrass form comes down to finding an inflection point, which exists on every irreducible plane cubic curve [17]. An algorithm for reduction to the Weierstrass form was implemented in the *algcurves* package for Maple [18].

Smoothness check or computation of a singular locus for a cubic surface can be used to model and render complex surfaces [19]. A cubic surface in $\mathbb{P}^3$ that is defined over a field K and contains a certain K-point is unirational over K, i.e., a set of K-points on this surface that is everywhere dense in the Zariski topology admits rational parameterization [20, 21]. Therefore, as compared to surfaces of higher degrees, cubic surfaces are more convenient to model [22, 23]. On the other hand, smooth real cubic surfaces with two connected components are unirational; however, they are irrational over the field of real numbers.

2. PRELIMINARY RESULTS

We assume that the projective space $\mathbb{P}^n$ with homogeneous coordinates $(x_0 : \cdots : x_n)$ is fixed in each dimension. The values of the homogeneous coordinates do not vanish simultaneously, and two sets of coordinates that differ in their common nonzero multipliers define the same point in $\mathbb{P}^n$. A square-free form $f(x_0, \ldots, x_n)$ defines a hypersurface at the points of which the form f vanishes. If, after some linear change of coordinates, the form depends on fewer variables, then it defines a cone in $\mathbb{P}^n$.

The Hessian matrix of a polynomial is a symmetric matrix of second-order partial derivatives. The entries of the Hessian matrix of a third-degree polynomial are linear functions. The determinant of the Hessian matrix is called the Hessian determinant (or simply the Hessian). The Hessian is defined by a hypersurface up to its nonzero multiplier, which allows us to speak of a Hessian of a hypersurface. The trace of the Hessian matrix is called the Laplacian. In Maple, the Hessian matrix, Hessian, and Laplacian are computed in the VectorCalculus package by using the Hessian command with the *determinant* option (for the Hessian) and Laplacian command (for the Laplacian).

The computational complexity of a matrix determinant has the same order of growth as matrix multiplication. There are algorithms that are asymptotically more efficient than the Gaussian method [24, 25]. The determinant is easy to compute, e.g., over the field of rational numbers or the field of rational functions in one variable with rational coefficients. However, for matrices over a ring of polynomials in many variables, generally speaking, the determinant cannot be computed in a time bounded by a polynomial expression in the number of variables. In this case, to check that the determinant does not vanish identically, the Schwartz–Zippel lemma [26] is usually employed.

Lemma 1 (Schwartz–Zippel). *Suppose that we have a polynomial $f(x_0, \ldots, x_n)$ of a positive degree $d > 0$ over a certain field. For any finite set S of elements of this field and for independent random variables $\xi_0, \ldots, \xi_n$ uniformly distributed on the set S, the probability for the polynomial to vanish $f(\xi_0, \ldots, \xi_n) = 0$ does not exceed $\frac{d}{|S|}$.*

For a cone in $\mathbb{P}^n$, the Hessian vanishes because the rows of the Hessian matrix are linearly dependent over a field of coefficients. However, the Hessian can also vanish when the rows of the Hessian matrix are linearly dependent only over a field of rational functions [27]. In 1876, P. Gordan and M. Noether showed that this condition holds for a cubic form $x_0x_3^2 + x_1x_3x_4 + x_2x_4^2$ that defines a non-conical hypersurface in $\mathbb{P}^4$.

Ternary forms with identically zero Hessians were considered by P.V. Bibikov [28]. In this paper, we confine ourselves to cubic forms in many variables.

If a hypersurface X is singular, then a set of singular points on X constitutes a subvariety called the singular locus. We consider hypersurfaces the singular loci of which contain sufficiently large linear subspaces.

By $E_0, \ldots, E_n$, we denote the points in $\mathbb{P}^n$ at which the homogeneous coordinates of a point E_k, except for the kth coordinate, are zero.

Theorem 1. *Suppose that we have a cubic form $f(x_0,\ldots,x_n)$ that defines a singular projective hypersurface $X \in \mathbb{P}^n$. If the linear coordinate subspace given by the equations $x_{m+1} = 0, \ldots, x_n = 0$ is embedded in the singular locus on X, then*

$$f = \sum_{k=0}^{m} x_k q_k(x_{m+1},\ldots,x_n) + c(x_{m+1},\ldots,x_n),$$

where q_k are certain quadratic forms and c is a cubic form.

Proof. Suppose that the form f contains a monomial $x_k^2 x_j$ for $k \le m$ and a certain j is possibly equal to k. Then, the partial derivative $\frac{\partial f}{\partial x_j}$ is nonzero at the point E_k, which contradicts the condition that E_k is a singular point. Therefore, for each $k \le m$, the form f does not contain monomials $x_k^2 x_j$ and, in particular, monomials x_k^3.

Suppose that the form f contains a monomial $x_k x_\ell x_j$ for some subscripts that satisfy the conditions $k < \ell \le m$, $j \ne k$, and $j \ne \ell$. Then, the partial derivative $\frac{\partial f}{\partial x_j}$ is nonzero at the point where only the kth and ℓth homogeneous coordinates are nonzero. However, by condition, this point is singular. This contradiction proves that there is no such monomial.

The existence of a low-dimensional singular locus does not imply that the Hessian vanishes. For instance, a Whitney umbrella defined by the form $x_0x_2^2 + x_1x_3^2$ represents a surface whose singular locus contains a straight line. This line is defined by two equations $x_2 = 0$ and $x_3 = 0$. However, the Hessian of this form is $16x_2^2x_3^2$. Another form $x_0x_3^2 + x_1x_4^2 + x_2x_5^2$ defines a hypersurface in $\mathbb{P}^5$ the singular locus of which contains a plane. The Hessian of this form is $-64x_3^2x_4^2x_5^2$.

Generally [29], for a set of forms whose coefficients depend on parameters $\xi_0, \ldots, \xi_m$, a certain property $\Phi(\xi_0,\ldots,\xi_m)$ holds for almost every form from this set if there is a polynomial $p(\xi_0,\ldots,\xi_m)$ that does not vanish identically but vanishes ($p(\xi_0,\ldots,\xi_m) = 0$) for each admissible set of its parameter values $\xi_0, \ldots, \xi_m$ if the property $\Phi(\xi_0,\ldots,\xi_m)$ does not hold. The property that holds for almost every set of parameter values does not hold only on a nowhere dense set of measure zero. However, in the general case, the set of the subsets of the parameter values on which the property Φ does not hold can be a proper subset of the set of zeros of the polynomial p.

For a matrix A, we denote its transpose by A^T.

Theorem 2. *Suppose that we have an odd number $n = 2m + 1$. For almost every cubic form $f(x_0,\ldots,x_n)$ that defines a projective hypersurface in $\mathbb{P}^n$ the singular locus of which contains a linear subspace given by the equations $x_{m+1} = 0, \ldots, x_n = 0$, the Hessian of the form f is a polynomial of degree $n + 1$ that does not depend on the variables $x_0, \ldots, x_m$.*

Proof. Theorem 1 implies that the Hessian matrix of the form f is a block matrix

$$H = \begin{pmatrix} 0 & A \\ A^T & B \end{pmatrix},$$

where A and B are square matrices of order $m + 1$ with the entries of A being linear forms that do not depend on $x_0, \ldots, x_m$. The Hessian of the form f, which is equal to $\det H$, does not depend on the block B and depends only on A.

The form $x_0x_{m+1}^2 + \cdots + x_mx_n^2$ satisfies the conditions of the theorem and its Hessian is $(-4)^{m+1}x_{m+1}^2 \cdots x_n^2$. According to Theorem 1, all cubic forms under consideration are parameterized by sets of coefficients of quadratic forms q_k and cubic form c. Therefore, there is a polynomial p in these coefficients the value of which is equal to the coefficient at the monomial $x_{m+1}^2 \cdots x_n^2$ of the Hessian of the corresponding form f with the degree of the polynomial p being positive. Hence, for almost every form f under consideration, the Hessian contains the monomial $x_{m+1}^2 \cdots x_n^2$ of degree $n + 1$. However, it does not contain monomials of higher degrees.

Recall the following sufficient condition for the Hessian of a cubic form to vanish [27].

Theorem 3 [27]. *Suppose that we have a cubic form $f(x_0,\ldots,x_n)$ that defines a projective hypersurface $X \in \mathbb{P}^n$. If, for certain $m > \frac{1}{2}(n-1)$, the linear subspace defined by the equations $x_{m+1} = 0, \ldots, x_n = 0$ is embedded in the singular locus on X, then the Hessian of the form f vanishes identically.*

Proof. According to Theorem 1, the first rows and columns of the Hessian matrix of f contains a zero submatrix of order $m + 1$. The order of the Hessian matrix is $n + 1$. Under the condition $m + 1 > \frac{1}{2}(n+1)$, the determinant of the Hessian matrix vanishes identically.

3. MAIN RESULTS

Theorem 4. *There is a probabilistic polynomial time algorithm that, as input, receives a rational number $\varepsilon > 0$, integer n, and cubic form $f(x_0,\ldots,x_n)$ given by a list of coefficients. If the form f defines a hypersurface in $\mathbb{P}^n$ the singular locus of which contains a linear subspace* *of a dimension higher than* $\frac{1}{2}n$, *then the input is always rejected. However, for almost every cubic form f, the input is accepted with a probability exceeding* $1 - \varepsilon$.

Proof. At the first step, the Hessian matrix of the form *f* is computed. At the second step, the probabilistic algorithm based on the Schwartz–Zippel lemma [26] checks that the Hessian of the form *f* does not vanish identically. If the singular locus of the hypersurface is sufficiently large, then, by virtue of Theorem 3, the Hessian vanishes for any estimate of the variables. The algorithm evaluates the Hessian on randomly and independently selected (on the interval from zero to $\lceil (n+1)/\varepsilon \rceil$) integer estimates of the variables $x_0, \ldots, x_n$. If the resulting value is zero, then the input is rejected; otherwise, the input is accepted.

Remark. A matrix of order $n + 1$ the entries of which are linear forms in variables $x_0, \ldots, x_n$ contains at most $(n+1)^3$ numerical coefficients. Using the symmetry of the Hessian matrix, we can store fewer numbers in memory. That is why the algorithm from Theorem 4 performs $O(n^3)$ operations on the field of coefficients. The algorithm is based on evaluating the determinant of the matrix over this field. If the cubic form *f* is defined over the field of rational numbers or over a finite extension of this field, then this probabilistic algorithm has polynomial bit complexity.

Theorem 5. *There is a probabilistic polynomial time algorithm that, as input, receives a rational number* $\varepsilon > 0$, *odd integer* $n = 2m + 1$, *and cubic form* $f(x_0, \ldots, x_n)$ *given by a list of coefficients. If the form f defines a hypersurface in* $\mathbb{P}^n$ *the singular locus of which contains a certain m-dimensional linear subspace, then the input is always rejected. However, for almost every cubic form f, the input is accepted with a probability exceeding* $1 - \varepsilon$.

Proof. First, the Hessian matrix of the form *f* is computed. Then, the probabilistic algorithm based on the Schwartz–Zippel lemma [26] checks the linear independence of the gradient of the Hessian of *f* at randomly selected points.

Suppose that $\xi_0 = 1$. The algorithm selects random estimates $\xi_1, \ldots, \xi_n$ for the variables $x_1, \ldots, x_n$, which, in turn, are uniformly and independently selected from a set of integers on the interval from zero to $\lceil n(n+1)/\varepsilon \rceil$. For $\xi_0 = 1$, these are homogeneous coordinates of a certain point in $\mathbb{P}^n$. To compute a partial derivative with respect to x_k at a selected point, we perform the substitution

$$x_j = \begin{cases} \xi_j, & j \neq k, \\ \xi_k + y, & j = k, \end{cases}$$

and evaluate the Hessian over a field of rational functions in one variable y. The first partial derivative of the Hessian with respect to the variable x_k is equal to the coefficient at the linear term in the variable y. Thus, the gradient at the selected point is computed. All computations are repeated for $n + 1$ independently selected points. The computations of the partial derivatives with respect to different variables at different points can be carried out in parallel.

Then, the determinant of the matrix of order $n + 1$, which consists of the first partial derivatives of the Hessian at $n + 1$ points, is computed. If the Hessian does not depend on any variable, then this determinant vanishes identically. Thus, if it vanishes in the process of computation, then the input is rejected; otherwise, the input is accepted.

The Hessian of the form *f* is a polynomial in variables $x_0, \ldots, x_n$ the degree of which does not exceed $n + 1$. The first partial derivative of the Hessian is a homogeneous polynomial of a degree not exceeding n. The determinant composed of these derivatives at $n + 1$ points has a degree not exceeding $n(n+1)$. According to the Schwartz–Zippel lemma, the probability for this polynomial to vanish (if it does not vanish identically) is less than ε.

Having confined ourselves to orthogonal transformations of coordinates, we can use both the Hessian and the Laplacian. Let us consider a hypersurface in $\mathbb{P}^n$ the singular locus of which can be zero-dimensional but contains points $E_0, \ldots, E_n$ at which the homogeneous coordinates of the point E_k (except for the *k*th coordinate) are zero. An example is the Cayley surface defined by the form $x_0x_1x_2 + x_0x_1x_3 + x_0x_2x_3 + x_1x_2x_3$.

Theorem 6. *There is a polynomial time algorithm that, as input, receives an integer n and cubic form* $f(x_0, \ldots, x_n)$ *given by a list of coefficients. If the form f defines a singular hypersurface in* $\mathbb{P}^n$ *the singular locus of which contains an image of points* $E_0, \ldots, E_n$ *under a certain orthogonal transformation, then the input is always rejected. However, for almost every cubic form f, the input is accepted.*

Proof. The Laplacian (the trace of a Hessian matrix) is invariant to orthogonal transformations of coordinates. Therefore, without loss of generality, we can assume that the form *f* defines a hypersurface with singular points $E_0, \ldots, E_n$. Following the proof of Theorem 1, we find that the form *f* is multilinear, i.e., no variable occurs in the second or third degree. Hence, the Laplacian vanishes identically. In contrast, for almost every cubic form, the Laplacian is equal to the linear form.

The algorithm rejects the input if the Laplacian vanishes identically; otherwise, the input is accepted.

4. DISCUSSION AND CONCLUSIONS

We have considered a sufficient condition for the absence of a sufficiently large singular locus on a cubic hypersurface, which can be checked in polynomial time by a probabilistic algorithm. In the general case, it does not allow us to conclude about the smoothness of this hypersurface. Moreover, with the smoothness being equivalent to the nonzero discriminant (irreducible polynomial of degree $(n+1)2^n$ in the coefficients of a cubic form in $n + 1$ variables), this property generally cannot be expressed in terms of a polynomial of a lower degree. The degree of the Hessian of this cubic form does not exceed $n + 1$. Thus, the method considered in this paper does not recognize singular loci of small dimensions. In addition, no deterministic algorithm for checking whether the Hessian vanishes has yet been reported.

In the probabilistic algorithms from Theorems 4 and 5, the boundaries of the segment from which the estimates of the variables are selected can be expanded in such a way that random numbers are in one-to-one correspondence to sets of random bits. In this case, the probability of error does not increase.

The algorithms from Theorems 4–6 are easy to transform into generic algorithms that either accept the input or issue a warning about impossibility of exact solution; however, they never reject the input or make errors. Indeed, the general hypersurface is smooth, while hypersurfaces with singular points are defined by forms with vanishing discriminants. On the other hand, when using a pseudo-random sequence instead of random bits, the number of inputs that are not accepted but result in a warning about infeasibility of computations by the generic algorithm can be significantly reduced by increasing the length of the segment on which random numbers are selected or by rerunning the test.

ACKNOWLEDGMENTS

We are grateful to Mark Spivakovsky for his participation in discussion of this work, as well as to S.A. Abramov and A.B. Batkhin for their helpful remarks.

REFERENCES

1. Alaev, P.E. and Selivanov, V.L., Fields of algebraic numbers computable in polynomial time. I, *Algebra Logic,* 2020, vol. 58, pp. 447–469. https://doi.org/10.1007/s10469-020-09565-0
2. Malaschonok, G.I., MathPartner computer algebra, *Program. Comput. Software,* 2017, vol. 43, no. 2, pp. 112–118. https://doi.org/10.1134/S0361768817020086
3. Neumann, E. and Pauly, A., A topological view on algebraic computation models, *J. Complexity,* 2018, vol. 44, pp. 1–22. https://doi.org/10.1016/j.jco.2017.08.003
4. Abramov, S.A., *Lektsii o slozhnosti algoritmov* (Lectures on Complexity of Algorithms), Moscow: Mosk. Tsentr Nepreryvnogo Mat. Obraz., 2012.
5. Rybalov, A.N., Generic amplification of recursively enumerable sets, *Algebra Logic,* 2018, vol. 57, no. 4, pp. 289–294. https://doi.org/10.1007/s10469-018-9500-y
6. Rybalov, A.N., On generic NP-completeness of the problem of Boolean circuits satisfiability, *Prikl. Diskretnaya Mat.,* 2020, no. 47, pp. 101–107. https://doi.org/10.17223/20710410/47/8
7. Seliverstov, A.V., On binary solutions of systems to equations, *Prikl. Diskretnaya Mat.,* 2019, no. 45, pp. 26–32. https://doi.org/10.17223/20710410/45/3
8. Eder, C. and Faugére, J.-C., A survey on signature-based algorithms for computing Gröbner bases, *J. Symbolic Comput.,* 2017, vol. 80, no. 3, pp. 719–784. https://doi.org/10.1016/j.jsc.2016.07.031
9. Yanovich, D.A., Computation of involutive and Gröbner bases using the tableau representation of polynomials, *Program. Comput. Software,* 2020, vol. 46, no. 2, pp. 162–166. https://doi.org/10.1134/S0361768820020115
10. Brzostowski, S., Krasiński, T., and Walewska, J., Arnold's problem on monotonicity of the Newton number for surface singularities, *J. Math. Soc. Jpn.,* 2019, vol. 71, no. 4, pp. 1257–1268. https://doi.org/10.2969/jmsj/78557855
11. Bryuno, A.D., The asymptotic behavior of solutions of nonlinear systems of differential equations, *Sov. Math. Dokl.,* 1962, vol. **3**, pp. 464–467.
12. Bryuno, A.D., Algorithms for solving an algebraic equation, *Program. Comput. Software,* 2018, vol. 44, no. 6, pp. 533–545. https://doi.org/10.1134/S0361768819100013
13. Bryuno, A.D., On the parametrization of an algebraic curve, *Math. Notes,* 2019, vol. 106, no. 6, pp. 885–893. https://doi.org/10.1134/S0001434619110233
14. Antipova, I.A., Mikhalkin, E.N., and Tsikh, A.K., Singular points of complex algebraic hypersurfaces, *J. Sib. Fed. Univ. Math. Phys.,* 2018, vol. 11, no. 6, pp. 670–679. https://doi.org/10.17516/1997-1397-2018-11-6-670-679
15. Gel'fand, I.M., Zelevinskii, A.V., and Kapranov, M.M., Discriminants of polynomials in many variables, *Funct. Anal. Appl.,* 1990, vol. 24, no. 1, pp. 1–4. https://doi.org/10.1007/BF01077912
16. Seliverstov, A.V., On tangent lines to affine hypersurfaces, *Vestn. Udmurt. Univ. Mat. Mekh. Komp'yut. Nauki,* 2017, vol. **27**, no. 2, pp. 248–256. https://doi.org/10.20537/vm170208
17. Rubanov, L.I. and Seliverstov, A.V., Projective-invariant description of a meandering river, *J. Commun. Technol. Electron.,* 2017, vol. 62, no. 6, pp. 663–668. https://doi.org/10.1134/S1064226917060201
18. van Hoeij, M., An algorithm for computing the Weierstrass normal form, *Proc. Int. Symp. Symbolic and Algebraic Computation (ISSAC)*, Levelt, A.H.M., Ed., New York: ACM Press, 1995, pp. 90–95.
19. Slyadnev, S.E. and Turlapov, V.E., Simplification of CAD models by automatic recognition and suppression of blend chains, *Program. Comput. Software,* 2020, vol. 46, no. 3, pp. 233–243. https://doi.org/10.1134/S0361768820030081
20. Segre, B., A note on arithmetical properties of cubic surfaces, *J. London Math. Soc.,* 1943, vol. 18, pp. 24–31.

21. Kollár, J., Unirationality of cubic hypersurfaces, *J. Inst. Math. Jussieu,* 2002, vol. 1, no. 3, pp. 467–476. https://doi.org/10.1017/S1474748002000117

22. Polo-Blanco, I. and Top, J., A remark on parameterizing nonsingular cubic surfaces, *Comput.-Aided Geom. Des.,* 2009, vol. 26, no. 8, pp. 842–849. https://doi.org/10.1016/j.cagd.2009.06.001

23. González-Sánchez, J. and Polo-Blanco, I., Construction algorithms for rational cubic surfaces, *J. Symbolic Comput.,* 2017, vol. 79, pp. 309–326. https://doi.org/10.1016/j.jsc.2016.02.010

24. Pan, V.Ya., Fast matrix multiplication and its algebraic neighbourhood, *Sb.: Math.,* 2017, vol. 208, no. 11, pp. 1661–1704. https://doi.org/10.1070/SM8833

25. Malaschonok, G., Recursive matrix algorithms, distributed dynamic control, scaling, stability, *Proc. Workshop Computer Science and Information Technologies (CSIT),* Yerevan, Armenia, 2019, pp. 112–115. https://doi.org/10.1109/CSITechnol.2019.8895255

26. Schwartz, J.T., Fast probabilistic algorithms for verification of polynomial identities, *J. ACM,* 1980, vol. 27, no. 4, pp. 701–717. https://doi.org/10.1145/322217.322225

27. Gondim, R. and Russo, F., On cubic hypersurfaces with vanishing hessian, *J. Pure Appl. Algebra,* 2015, vol. 219, no. 4, pp. 779–806. https://doi.org/10.1016/j.jpaa.2014.04.030

28. Bibikov, P.V., Classification of ternary forms with zero hessian, *Russ. Math.,* 2011, vol. 55, no. 9, pp. 83–85. https://doi.org/10.3103/S1066369X11090118

29. Seliverstov, A.V., Symmetric matrices whose entries are linear functions, *Comput. Math. Math. Phys.,* 2020, vol. 60, pp. 102–108. https://doi.org/10.1134/S0965542520010121

Translated by Yu. Kornienko