Computing unimodular matrices of power transformations

Alexander D. Bruno[1], Alijon A. Azimov[2]
abruno@keldysh.ru
Azimov_Alijon_Akhmadovich@mail.ru

[1]*Keldysh Institute of Applied Mathematics of RAS*
[2]*Samarkand State University after Sh. Rashidov*

PCA-2023, April 17–22, 2023

Talk outlook

1. Introduction

2. Euclidean algorithm and continued fraction

3. Euler's algorithm and a generalization of continued fraction

4. Solution to Problem 1

5. Programs for calculating continued fractions

6. Implementation of Euler's algorithm and solution of Problem 1

7. Power transformations

Abstract

An algorithm for solving the following problem is described. Let $m < n$ integer vectors in the n-dimensional real space be given. Their linear span forms a linear subspace L in $\mathbb{R}^n$. It is required to find a unimodular matrix such that the linear transformation defined by it takes the subspace L into a coordinate subspace. Computer programs, implementing the proposed algorithms, and the power transformations, for which they are designed, are described.

1. Introduction (1)

Recall that a square matrix is said to be ***unimodular*** if all its elements are integers and its determinant equals ± 1. Its inverse and transpose are also unimodular.

We will write vectors as row vectors $A = (a_1, \ldots, a_n)$,and $[a]$ is the integer part of the real number a.

1. Introduction (2)

Problem 1

Let m, $(m < n)$ integer vectors $A_1, \dots, A_m$ be given in the n-dimensional real space $\mathbb{R}^n$. Their linear span

$$L = \left\{ X = \sum_{j=1}^{m} \lambda_j A_j,\, \lambda_j \in \mathbb{R}, \quad j = 1, \dots, m \right\} \tag{1}$$

forms a linear subspace in $\mathbb{R}^n$. It is required to find a unimodular matrix α such that the transformation $X\alpha = Y$ takes L to the coordinate subspace

$$M = \{Y : y_1 = \dots = y_{n-l} = 0\},$$

where $l = \dim L$.

1. Introduction (3)

In this talk, we give an algorithm for solving this problem and provide its implementations in computer algebra systems [Bruno, Azimov, 2023]. This is a generalization of the continued fraction algorithm [Khinchin, 1997], which we recall in Section 2. In Section 3, we describe the Euler algorithm [Euler, 1785], which generalizes the Euclidean algorithm (i.e., the continued fraction algorithm) to the n-dimensional integer vector. In Section 4 we describe a solution of Problem 1 and in Sections 5 and 6, we present programs corresponding to Sections 2, 3, and 4. In Section 7 we consider power transformations, for the calculation of the unimodular matrices of which, all these algorithms are developed.

1. Introduction

2. **Euclidean algorithm and continued fraction**

3. Euler's algorithm and a generalization of continued fraction

4. Solution to Problem 1

5. Programs for calculating continued fractions

6. Implementation of Euler's algorithm and solution of Problem 1

7. Power transformations

2. Euclidean algorithm and continued fraction (1)

Problem 2

Let a_1 and a_2 two positive integers. We want to find their greatest common divisor (GCD).

2. Euclidean algorithm and continued fraction (2)

For that *Euclidean algorithm* can be used.

Let $a_1 \geqslant a_2$, divide a_1 by a_2 with remainder:

$$a_1 = b_1 \cdot a_2 + a_3, \tag{2}$$

where $b_1 = [a_1/a_2]$ and a_3 are integers and $0 \leqslant a_3 < a_2$.

- If $a_3 = 0$, then the GCD is a_2.
- If $a_3 \neq 0$, then we divide a_2 by a_3 with remainder:

$$a_2 = b_2 \cdot a_3 + a_4, \tag{3}$$

  where $b_2 = [a_2/a_3]$ and $0 \leqslant a_4 < a_3$.
- If $a_4 = 0$, then the GCD is a_3.
- If $a_4 \neq 0$, then we continue this procedure until we obtain the zero remainder $a_{k+1} = 0$.

2. Euclidean algorithm and continued fraction (3)

Then, the GCD is a_k. This procedure can be written as the continued fraction

$$\frac{a_1}{a_2} = b_1 + \cfrac{1}{b_2 + \cfrac{1}{b_3 + \cfrac{1}{\ldots + \cfrac{1}{b_{k-1}}}}}. \tag{4}$$

This procedure is applicable to any real number β and gives, generally speaking, an infinite expansion. It is finite only for rational numbers $\beta = a_1/a_2$ For quadratic irrationalities β, it is periodic [**khinchin**]. If we discard in the continued fraction (4) the tail starting with b_{l+1} and collapse the resulting continued fraction into a rational number, then it is called a ***convergent***.

2. Euclidean algorithm and continued fraction (4)

Problem 3

Let a_1 and a_2 be two positive integers. It is required to find a unimodular matrix α such that $(a_1, a_2)\,\alpha = (a_k,0)$ or $(0, a_k)$, where $a_k > 0$ is an integer.

Division with remainder (2) can be written as multiplication by the matrix $(a_1, a_2)\begin{pmatrix} 1 & 0 \\ -b_1 & 1 \end{pmatrix} = (a_3, a_2)$ or $(a_1, a_2)\,\beta_1 = (a_3, a_2)$, where $\beta_1 = \begin{pmatrix} 1 & 0 \\ -b_1 & 1 \end{pmatrix}$, and division with remainder (3) is represented by $(a_3, a_2)\begin{pmatrix} 1 & -b_2 \\ 0 & 1 \end{pmatrix} = (a_3, a_4)$ or $(a_3, a_2)\,\beta_2 = (a_3, a_4)$, where $\beta_2 = \begin{pmatrix} 1 & -b_2 \\ 0 & 1 \end{pmatrix}$.

2. Euclidean algorithm and continued fraction (5)

The last step of the Euclidean algorithm is either

$$(a_k, a_{k-1}) \begin{pmatrix} 1 & -b_{k-1} \\ 0 & 1 \end{pmatrix} = (a_k, 0)$$

or

$$(a_{k-1}, a_k) \begin{pmatrix} 1 & 0 \\ -b_{k-1} & 1 \end{pmatrix} = (0, a_k).$$

2. Euclidean algorithm and continued fraction (6)

Therefore, the desired matrix is

$$\alpha = \beta_1 \beta_2 \cdots \beta_{k-1},$$

where

$$\beta_j = \begin{pmatrix} 1 & 0 \\ -b_j & 1 \end{pmatrix}, \tag{5}$$

if j is odd, and

$$\beta_j = \begin{pmatrix} 1 & -b_j \\ 0 & 1 \end{pmatrix}, \tag{6}$$

if j is even.

Since all matrices β_j are unimodular, their product α is unimodular as well, and it provides a solution to Problem 3.

2. Euclidean algorithm and continued fraction (7)

Note that

$$\alpha^{-1} = \beta_{k-1}^{-1}\beta_{k-2}^{-1}\cdots\beta_2^{-1}\beta_1^{-1}$$

and, according to (5) and (6) $\beta_j^{-1} = \begin{pmatrix} 1 & 0 \\ b_j & 1 \end{pmatrix}$ or $\begin{pmatrix} 1 & b_j \\ 0 & 1 \end{pmatrix}$, i.e. it consists of nonnegative elements. Hence, all elements in the matrix α^{-1} are nonnegative.

Example (1)

Let $a_1 = 17$ and $a_2 = 5$.
Then $b_1 = [17/5] = 3$, $a_3 = 2$, $\beta_1 = \begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix}$,
$b_2 = \left[\frac{5}{2}\right] = 2,\ a_4 = 1,\ \beta_2 = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}$,
$b_3 = \left[\frac{2}{1}\right] = 2,\ a_5 = 0,\ \beta_3 = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}$

The matrix $\alpha = \beta_1\beta_2\beta_3 = \begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix}\begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} = \begin{pmatrix} 5 & -2 \\ -17 & 7 \end{pmatrix}$.

Example (2)

Expand the ratio in continued fraction

$$\frac{17}{5} = 3 + \cfrac{1}{2 + \cfrac{1}{2}},$$

and find that the last convergent is $3 + 1/2 = 7/2$. Therefore, the second column in the matrix α consists of the numbers -2 and 7. $\square$

Euclidean algorithm and continued fraction (cont) (1)

Here we described the solution to Problem 3 for the case a_1, $a_2 > 0$. If a_1, $a_2 < 0$, then we should take the matrix α for the vector $(-a_1, -a_2)$.

If $a_1 \cdot a_2 < 0$, then we should take the matrix $\alpha = \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix}$ for the vector $(|a_1|, |a_2|)$. Then, the matrix

$$\alpha = \begin{pmatrix} \alpha_{11} \operatorname{sign} a_1 & \alpha_{12} \operatorname{sign} a_2 \\ \alpha_{21} \operatorname{sign} a_1 & \alpha_{22} \operatorname{sign} a_2 \end{pmatrix}$$

is unimodular and makes one of the coordinates of the vector (a_1, a_2) equal to zero.

Euclidean algorithm and continued fraction (cont) (2)

Here we assumed that $|a_1| \geqslant |a_2|$. Otherwise, we should first rearrange the coordinates:

$$(a_1, a_2) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = (a_2, a_1) \, .$$

A similar exposition can be found in [Bruno, 1989, Part I, Ch. I, § 1, Sect. 1.9].

1. Introduction
2. Euclidean algorithm and continued fraction
3. **Euler's algorithm and a generalization of continued fraction**
4. Solution to Problem 1
5. Programs for calculating continued fractions
6. Implementation of Euler's algorithm and solution of Problem 1
7. Power transformations

3. Euler's algorithm and a generalization of continued fraction (1)

Problem 4

Let an n-dimensional integer vector $A = (a_1, a_2, \ldots, a_n)$ be given. Find an n-dimensional unimodular matrix α such that the vector $A\alpha = C = (c_1, \ldots, c_n)$ contains only one nonzero component c_n.

Euler proposed the following algorithm for solving this problem [Euler, 1785]. Suppose for the time being that all components of vector A are nonnegative.

Using the permutation $A\alpha_0 = (\widetilde{a}_1, \widetilde{a}_2, \ldots, \widetilde{a}_n)$ arrange its components in nondecreasing order $\widetilde{a}_j \leq \widetilde{a}_{j+1}$, $j = 1, \ldots, n-1$. Here α_0 is the unimodular matrix of the permutation.

Let $\widetilde{a}_k$ be the least number among $\widetilde{a}_j$ that is distinct from zero.

3. Euler's algorithm and a generalization of continued fraction (2)

Let $b_j = [\widetilde{a}_j/\widetilde{a}_k]$, $j = 1, \dots, n$. Here $b_1 = \dots = b_{k-1} = 0$, $b_k = 1$. Make the transformation

$$d_j = \widetilde{a}_j - b_j\widetilde{a}_k,\ 1 \leqslant j \leqslant n,\ j \neq k, d_k = \widetilde{a}_k.$$

It is associated with the unimodular matrix α_1, the diagonal of which consists of ones, and the k-th row is

$$0,0,\dots,0,1,-b_{k+1},\dots,-b_n,\ \text{i.e.}\ \tilde{A}\alpha_1 = D = (d_1,\dots,d_n).$$

Now arrange the components of the vector D in non-decreasing order using the unimodular permutation matrix β_0 so that $D\beta_0 = \tilde{D} = \left(0,\dots,0,\widetilde{d}_k,\dots,\widetilde{d}_n\right)$, where $\widetilde{d}_j \leqslant \widetilde{d}_{j+1}$.

3. Euler's algorithm and a generalization of continued fraction (3)

Let $\widetilde{d}_l$ be the least of $\widetilde{d}_j$, distinct from zero, and let $e_j = \left[\widetilde{d}_j / \widetilde{d}_l\right]$, $j = 1, \ldots, n$. Make the transformation

$$f_j = \widetilde{d}_j - e_j \widetilde{d}_l, \quad 1 \leqslant j \leqslant n, \quad j \neq l, \quad f_l = \widetilde{d}_l,$$

and so on.

At each step, the maximum of the components of the vector decreases and it is the n-th component. Therefore, in a finite number of steps we obtain a vector with the only (last) nonzero component. This component equals the GCD of all original components $a_1, \ldots, a_n$.

3. Euler's algorithm and a generalization of continued fraction (4)

Each step involves a permutation matrix and a triangular matrix with the unit diagonal:

$$A\alpha_0\alpha_1\beta_0\beta_1\gamma_0\gamma_1\ldots\omega_0\omega_1 = A\alpha = C = (0,\ldots,0,c_n).$$

The matrix

$$\alpha = \alpha_0\alpha_1\beta_0\beta_1\gamma_0\gamma_1\cdots\omega_0\omega_1$$

is a solution of Problem 4.

If not all components a_j of the original vector A have the same sign, then we first arrange them in non-decreasing order of their moduli $|\widetilde{a}_j| \leqslant |\widetilde{a}_{j+1}|$ and set $b_j = [|\widetilde{a}_j| \,/\, |\widetilde{a}_k|] \operatorname{sign}\widetilde{a}_j \operatorname{sign}\widetilde{a}_k$.

3. Euler's algorithm and a generalization of continued fraction (5)

Euler's algorithm generalizes the continued fraction algorithm only for integer vectors. Such a generalization for arbitrary real vectors was sought by all major mathematicians of the 19th century, but without success.

Such a generalization of the continued fraction algorithm for the n-dimensional vector was proposed in [Bruno, 2019]. It gives a sequence of best approximations, and it is periodic if all the components of the original vector are roots of a polynomial of degree n with integer coefficients.

4. Solution to Problem 1 (1)

Let integer vectors

$$\begin{aligned} A_1 &= (a_{11}, a_{12}, \ldots, a_{1n}), \\ A_2 &= (a_{21}, a_{22}, \ldots, a_{2n}), \\ &\ldots \\ A_m &= (a_{m1}, a_{m2}, \ldots, a_{mn}) \end{aligned} \tag{7}$$

$(m < n)$ and the linear space (1) be given.

First, we check if there are identical vectors among them. If there are any, we discard duplicates and leave only one of them. Now, we are sure that all vectors (7) are different. Apply Euler's algorithm to the vector A_1, i.e., calculate the matrix α such that $A_1\alpha_0 = C_1 = c_n E_n$, where c_n is an integer and E_k is the k-th unit vector.

4. Solution to Problem 1 (2)

Let $A_j\alpha_0 = C_j = (c_{j1}, \ldots, c_{jn})$, $j = 2, \ldots, m$. Set $A_j^1 = (c_{j1}, \ldots, c_{jn-1})$, $j = 2, \ldots, m$. Apply Euler's algorithm to the $(n-1)$-dimensional vector A_2^1 to obtain $A_2^1\alpha_1 = C_2^1 = (0{,}0, \ldots, c_{n-1}^1)$, where α_1 is an $(n-1)$-dimensional square matrix. Let

$$A_j^1\alpha_1 = C_j^1 = (c_{j1}^1, \ldots, c_{jn-1}^1), \quad j = 3, \ldots, m.$$

Apply Euler's algorithm to the $(n-2)$-dimensional vector C_3^1, and so on. Finally, we obtain the sequence of matrices $\alpha_0, \alpha_1, \ldots, \alpha_{m-1}$ of decreasing size $n, n-1, \ldots, n-m+1$.

4. Solution to Problem 1 (3)

Form the block matrices

$$\beta_j = \begin{pmatrix} \alpha_j & 0 \\ 0 & I_{j+1} \end{pmatrix}, \quad j = 0, \dots, n-m,$$

of size n, where I_{j+1} are the identity matrices of size $j+1$. Set $\gamma = \beta_0 \beta_1 \cdots \beta_{m-1}$.

Then

$$A_j \gamma = (0,0,\dots,0, w_{j,n-j+1}, \dots, w_{j,n}) = W_j, \quad j = 1, \dots, m.$$

The matrix γ is a solution to Problem 1.

5. Programs for calculating continued fractions (1)

Algorithms for calculating continued fractions are implemented in many computer algebra systems. Here we describe the basic procedures in two computer algebra systems: in the proprietary system `Maple` and in the free system `sympy`.

The package `NumberTheory` in `Maple` [Thompson, 2016] makes it possible to expand rational, algebraic and transcendental numbers, and polynomials and elementary functions of one variable in continued fractions.

In `sympy` [Meurer (et al.), 2017], this functionality is implemented only for rational numbers or quadratic irrationalities. If we need to work with continued fractions for other irrationalities and transcendental numbers, the free system `Sage` [The Sage Developers, 2022] should be used.

5. Programs for calculating continued fractions (2)

Three basic procedures are sufficient for working with rational numbers in the form of continued fractions:

(1) transformation into continued fraction;

(2) obtaining elements of the continued fraction;

(3) obtaining rational approximations.

In `Maple`, these procedures are implemented by `ContinuedFraction`, `Term`, and `Convergent`.

In `sympy`, procedures (1) and (2) are implemented by `continued_fraction`, and procedure (3) by `continued_fraction_convergents`.

1. Introduction

2. Euclidean algorithm and continued fraction

3. Euler's algorithm and a generalization of continued fraction

4. Solution to Problem 1

5. Programs for calculating continued fractions

6. Implementation of Euler's algorithm and solution of Problem 1

7. Power transformations

6. Implementation of Euler's algorithm and solution of Problem 1 (1)

To implement Euler's algorithm described in Section 3, we developed a set of `Maple` procedures, which are presented below together with their descriptions. Note that an integer vector in `Maple` can be represented in two forms:

- a list of numbers in square brackets;
- a row vector (or column vector) of the package `LinearAlgebra` (`Vector[row]` or `Vector[column]`, respectively).

If the procedure name contains digit 2, then the input integer vector may be specified in any of these forms.

6. Implementation of Euler's algorithm and solution of Problem 1 (2)

The procedure *`MakePermute2`*:

Given the vector A, it constructs the permutation matrix. This procedure produces the ordered vector and the permutation matrix α_0. The order in which the components of the vector are arranged is specified by the parameter sorting; by default, the components are arranged in increasing order. At the beginning of its work, the procedure checks that the dimension of the vector A is greater than zero.

According to the algorithm described in Section 4, the solution of Problem 1 is implemented by the recursive ***procedure*** `UniSys`.

6. Implementation of Euler's algorithm and solution of Problem 1 (3)

The procedure gets at its input a set of integer vectors A_j $(j = 1, \ldots, m)$ in the form of the list `Vlst`. If the input list is empty (row 5) or if the number of vectors is greater than their dimensionality (row 8), or if the vectors are linearly dependent (rows 9–15), or if the vectors in the input set have different types (rows 16–24) or different dimension (rows 25–28), then the procedure `UniSys` terminates.

6. Implementation of Euler's algorithm and solution of Problem 1 (4)

If the list consists of a single vector, then the procedure `Unimodr2` is called, the matrix α is calculated, and the procedure terminates. Otherwise, the procedure `UniSys` is called again for the set of vectors A_j^1 $(j = 2, \ldots, m)$ from which the first j vector is removed, and the unimodular matrix α is applied. In this case, the dimension of the vectors A_j^1 decreases by one, and the matrix α is passed in the parameter `Uni` for the repeated call of the procedure. If the procedure `UniSys` terminates successfully, then it returns the resulting matrix γ that solves Problem 1.

1. Introduction

2. Euclidean algorithm and continued fraction

3. Euler's algorithm and a generalization of continued fraction

4. Solution to Problem 1

5. Programs for calculating continued fractions

6. Implementation of Euler's algorithm and solution of Problem 1

7. Power transformations

7. Power transformations (1)

Let the polynomial

$$f(X) = \sum f_Q X^Q \text{ over } Q \in \mathbf{S},$$

where $X = (x_1, \ldots, x_n) \in \mathbb{R}^n$ or $\mathbb{C}^n$, $X^Q = x_1^{q_1} \cdots x_n^{q_n}$, $Q = (q_1, \ldots, q_n) \in \mathbb{Z}^n, Q \geqslant 0, f_Q$ are constant coefficients from $\mathbb{R}$ or $\mathbb{C}$, $\mathbf{S} = \mathbf{S}(f)$ is the support of f, be given. Let $\mathcal{F}$ be the algebraic variety $f(X) = 0$ and the point $X = X^0 \in \mathcal{F}$.

If X^0 is a simple point, i.e., if at least one derivative $\partial f / \partial x_j$ is nonzero at X^0 then the implicit function theorem implies that the variety $\mathcal{F}$ in the neighborhood of X^0 is described by the equation

$$\Delta x_j = \varphi(\Delta x_1, \ldots, \Delta x_{j-1}, \Delta x_{j+1}, \ldots, \Delta x_n),$$

where $\Delta x_k = x_k - x_k^0$ and φ is a convergent series of its arguments.

7. Power transformations (2)

If X^0 is not a simple point, then, according to [Bruno, 2000; Bruno, Batkhin, 2012] we can seek the branches of the variety $\mathcal{F}$, passing through X^0 in the form of parametric expansions

$$\Delta x_j = \varphi_j(\xi_1, \ldots, \xi_{n-1}), \quad j = 1, \ldots, n,$$

where ξ_k are small parameters and φ_j — are converging power series. To this end the convex hull Γ of the support $\mathbf{S}$ in the space $\mathbb{R}^n$ is constructed.

Then, Γ is the polyhedron, the boundary $\partial\Gamma$ of which consists of (generalized) faces $\Gamma_j^{(d)}$ of dimension d, $0 \leq d < n$. Here j is the face index. Since all vertices $\Gamma_j^{(0)}$ of Γ are integer, each face $\Gamma_j^{(d)}$ has $n - d$ integer linearly independent normals $N_{j,1}^{(d)}, \ldots, N_{j,n-d}^{(d)} \in \mathbb{R}_*^n$ i.e., normals belonging to the space $\mathbb{R}_*^n$, which is dual of the space $\mathbb{R}^n$.

7. Power transformations (3)

In addition, each face $\Gamma_j^{(d)}$ is associated with the boundary set

$$D_j^{(d)} = \left\{ Q \in \mathbf{S} \cap \Gamma_j^{(d)} \right\},$$

and the truncated sum

$$\hat{f}_j^{(d)}(X) = \sum f_Q X^Q \text{ over } Q \in D_j^{(d)}. \tag{8}$$

Theorem 7.1 ([Bruno, 2000, Corollary in Chapter II, § 3], [Bruno, Batkhin, 2012, Theorem 3.1]).

For the face $\Gamma_j^{(d)}$ *there exists a power transformation*

$$\ln Y = \ln X \cdot \alpha,$$

where $\ln Y = (\ln y_1, \dots, \ln y_n)$ *and* $\ln X = (\ln x_1, \dots, \ln x_n)$ *with a unimodular matrix* α, *that takes the truncated sum* (8) *to a polynomial* g *of* d *variables, i.e.,*

$$\hat{f}_j^{(d)}(X) = Y^T g(y_1, \dots, y_d),$$

where $T = (t_1, \dots, t_n) \in \mathbb{Z}^n$.

7. Power transformations (5)

However in [Bruno, 2000; Bruno, Batkhin, 2012], it was not pointed out how the unimodular matrix α can be calculated. This is done in the current paper. In [Bruno, 1989, Part I, Ch. I, Section 1.9] it was made for $n = 2$. In [Bruno, Azimov, 2022; 2023] we describe software of these algorithms. It will be considered in our talk.

This approach works for differential equations as well. For a system of partial differential equations (PDE) it will be shown in talk ***"Asymptotic forms of solutions to a system of PDE"*** by A.D.Bruno and A.B.Batkhin.

References (1)

Bruno A. D. Local Methods in Nonlinear Differential Equations. Berlin – Heidelberg – New York – London – Paris – Tokyo: Springer–Verlag, 1989.

Bruno A. D. Power Geometry in Algebraic and Differential Equations. Amsterdam: Elsevier Science, 2000.

Bruno A. D. Computation of the Fundamental Units of Number Rings Using a Generalized Continued Fraction. // Programming and Computer Software. 2019. Vol. 45, no. 2. P. 37–50. DOI: 10.1134/S036176881902004X. URL: https://disk.yandex.ru/i/sCbG45VZGVY7wQ.

Bruno A. D., *Azimov A. A.* Computation of unimodular matrices. // Preprints of KIAM. Moscow, 2022. No. 46. P. 1–20. DOI: https://doi.org/10.20948/prepr-2022-46. (in Russian).

Bruno A. D., Azimov A. A. Computation of unimodular matrices of power transformations. // Programming and Computer Software. 2023. Vol. 49, no. 1. P. 32–41. DOI: 10.1134/S0361768823010036. URL: https://disk.yandex.ru/i/zXlMlHu91CcQiw.

Bruno A. D., Batkhin A. B. Resolution of an algebraic singularity by power geometry algorithms. // Programming and Computer Software. 2012. Vol. 38, no. 2. P. 57–72. DOI: 10.1134/S036176881202003X. URL: https://disk.yandex.ru/i/dmylwZQZxQ_wTw.

Euler L. De relatione inter ternas pluresve quantitates instituenda. //. Vol. 4. Opera Omnia, 1785. P. 136–145. (1). E591 (On the relation between three and more quantities which are to be instituted).

References (3)

Khinchin A. Y. Continued Fractions. Revised edition. NY: Dover Publications, 1997. 112 p. (in Russian: Moscow: Fizmatgiz, 1961).

Meurer A. [et al.]. SymPy: symbolic computing in Python. // PeerJ Computer Science. 2017. Vol. 3. e103. DOI: 10.7717/peerj-cs.103.

The Sage Developers. SageMath, the Sage Mathematics Software System (Version 9.7). 2022. https://www.sagemath.org.

Thompson I. Understanding Maple. Cambridge University Press, 2016. 228 p.

Thanks for your attention!