34. Gupta D., Gupta V. K., Chandra M. Performance Comparison of Different Affine Projection Algorithms for Noise Minimization from Speech Signals // International Journal of Future Generation Communication and Networking. 2017. Vol. 10, Issue 1. P. 261–270. doi: https://doi.org/10.14257/ijfgcn.2017.10.1.21
35. A statistical analysis of the affine projection algorithm for unity step size and autoregressive inputs / De Almeida S. J. M., Bermudez J. C. M., Bershad N. J., Costa M. H. // IEEE Transactions on Circuits and Systems I: Regular Papers. 2005. Vol. 52, Issue 7. P. 1394–1405. doi: https://doi.org/10.1109/tcsi.2005.851720
36. Wagner K. T., Doroslovacki M. I. Towards analytical convergence analysis of proportionate-type NLMS algorithms // 2008 IEEE International Conference on Acoustics, Speech and Signal Processing. 2008. doi: https://doi.org/10.1109/icassp.2008.4518487

UDC 519.852:519.876
DOI: 10.15587/1729-4061.2019.157521

ADVANCEMENT OF A LONG ARITHMETIC TECHNOLOGY IN THE CONSTRUCTION OF ALGORITHMS FOR STUDYING LINEAR SYSTEMS

V. Kudin
Doctor of Technical Sciences, Professor
Department of Intelligent and Information Systems*
E-mail: vol.kudin@gmail.com

V. Onotskyi
PhD, Assistant
Department of Computational Mathematics*
E-mail: vingar@ukr.net

A. Al-Ammouri
Doctor of Technical Sciences, Professor
Department of Information Analysis and Information Security
National Transport University
Mykhaila Omelianovycha-Pavlenka str., 1, Kyiv, Ukraine, 01010
E-mail: ammourilion@ukr.net

L. Shkvarchuk
Doctor of Economic Sciences, Professor
Department of Finance
Lviv Polytechnic National University
S. Bandery str., 12, Lviv, Ukraine, 79013
E-mail: S_Lyuda16@ukr.net

*Taras Shevchenko National University of Kyiv
Volodymyrska str., 60, Kyiv, Ukraine, 01033

Розвинуто застосування алгоритмів методу базисних матриць, які оснащені технологією довгої арифметики для покращення точності виконання основних операцій при дослідженні погано обумовлених лінійних систем, зокрема систем лінійних алгебраїчних рівнянь (СЛАР). Встановлення факту поганої обумовленості системи є досить трудомісткою обчислювальною процедурою. Закладено проведення контролю входження обчислень в стан некоректності та унеможливлення накопичення похибок обчислень, що є бажаною властивістю методів та алгоритмів розв'язання практичних задач.

В сучасних ЕОМ, як правило, використовуються стандарті типи цілих чисел, розмір яких не перевищує 64 байта. Було подолано це апаратне обмеження програмним шляхом, а саме, розробкою власного типу даних у вигляді спеціальної бібліотеки Longnum мовою C++ з використанням стандартної бібліотеки шаблонів STL(Standard Template Library). Програмна реалізація була розвинута на проведення обчислень за методами базисних матриць (МБМ) та Гауса, тобто використано довгу арифметику для моделей з раціональними елементами. Запропоновано алгоритми та комп'ютерну реалізацію методів типу Гауса та штучних базисних матриць (варіант методу базисних матриць) в середовищах Matlab та Visual C++ з використанням технології точних обчислень елементів методів, в першу чергу, для погано обумовлених систем різної розмірності. Розроблено бібліотеку Longnum з типами довгих цілих чисел (longint3) та раціональних чисел (longrat3) із чисельником та знаменником типу longint3. Арифметичні операції над довгими цілими числами реалізовано на основі сучасних методів: зокрема, методу Штрасена множення. Наведено результати обчислювального експерименту за згаданими методами, в якому тестові моделі систем генерувались, зокрема, на основі матриць Гільберта різної розмірності, які характеризуються як "незручні"

Ключові слова: метод базисних матриць, точні обчислення, погано обумовлена система лінійних рівнянь

1. Introduction

Mathematical modeling of processes of different nature is known to lead to the need to explore non-linear equations and systems of varying complexity (mathematical models). In many cases, they are solved by the introduction of certain simplifications in statements, transition, in particular, to difference analogues (discrete variant) and eventually to the systems of linear algebraic equations (SLAE) of different dimension, often with a square matrix of restrictions. The methods and algorithms of SLAE (as the basic ones) have become the subject-matter of research of many scientists. This, in turn, caused the development of (currently) many dozens of exact methods and, perhaps, hundreds of iterative methods. An exceptional position in the chain of the solution of an initial problem of modeling caused exceptional requirements for computing properties of the methods and algorithms for the solution of the SLAE. Natural complexity of solving such problems is manifested in incorrectness and ill-conditionality. The efforts of calculators are aimed to overcome (or weaken) the manifestation of ill-conditionality and incorrectness in the course of solving a problem and (accumulation of calculation errors). There appear new approaches to solving the known problems. According to the authors, one of the promising approaches is the development of the known methods (including the algorithms of the method of basic matrices), equipment with the technology of long arithmetic during studying ill-conditioned linear systems. Of course, the latest technology (of long arithmetic) involves the use of other technologies, which will be mentioned below, specifically, the introduction of a new data class. It is the successful combination of a series of new technologies of computation algorithms organization that can provide new positive possibilities.

The problem of modeling the processes of different nature is often described by a class of optimal problems (for extremum), in particularly, the SLAE. Precise methods for the solution of the SLAE in the number of the above problems are fundamental, since research into the initial more complex mathematical models “is reduced” (after some simplifications) to the analysis of such linear systems.

It is known that:

– most mathematical problem statements (in the first place, on studying the properties of processes) are inherently non-linear, that is they do not have any adequate representation in the class of linear models (matrix structures) – the systems of linear algebraic equations, inequalities, linear programming problems, etc.;

– important parameters of the process (in the process of simulation and simplifications) are mapped (“transfer”) to separate elements, rows, columns and blocks (“sub-matrices”) of the restriction matrix, etc., that is, undergo changes and clarifications. Therefore, taking into consideration the impact of the changes in a model as a result of specifications (without re-solving) is a desirable characteristic of the methods of modeling.

The existing property of ill-conditionality and incorrectness in the problem is at all stages of the simulation. Specifically, at the stage of a computer representation of a model – rounding, truncation, the limited length of the mantissa causes the accumulation of mistakes, errors, inaccuracies of the model representation, etc. This gives rise to inadequacy of the studied process and the model. It should be noted that in such situations, even minor quantitative inaccuracy in the representation of a model can often lead to significant deviation (error) of the solution. Equipment of algorithms with the technology that prevents the accumulation of errors at the stages of modeling, specifically, during the iterations of the algorithm of the method is the relevant direction of research.

2. Literature review and problem statement

Up to now, quite a lot of precise and iterative methods for solving systems of linear algebraic equations have been developed. Specifically, some of them are listed in papers [3, 4] and there exist many others. It may be stated that the problem of the development of a universal high-precision method for solving a wide class of linear problems has not been solved so far. This can be explained by the objective complexities of problems that arise, especially when modeling the processes of different nature. Mathematical models are quite often represented as great dimensionalities, are incorrect (sensitive to inaccuracies), ill-conditioned, etc. [1, 2]. Technological advances of today, such as increasing the digits of processors up to 64 and the volumes of different memory types, together with the increasing rate of performing operations, provide additional opportunities for representing long numbers. However, they do not solve the problem of the representation of numbers and implementation of operations for the numbers with more than 64 digits and it remains open. In this regard, for various kinds of ill-conditionality or the structure of the restriction matrix, there is a need to adapt the solution algorithm to ensure the computational quality [3–5].

Of course, among them, a special place is taken by algorithmic schemes that are based on the Gaussian method. It is known that the SLAE models can often be incorrect by their structural properties (natural complexity). One of the manifestations of incorrectness is the property of ill-conditionality (conditionality number $M_A = \|A\| \times \|A^{-1}\|$ takes large values that affect the errors of performing basic operations) [1–5]. It is possible to make sure that the information on the significance of the conditionality number (or its assessment) as a factor of control of calculation correctness currently remains in the scope of further research [3–5]. Specifically, the existence of the system conditionality control, accumulation of calculation errors when solving such inconvenient problems is a necessary and integral component of the computation process. From the point of view of the classical methods of the Gaussian type, for example, the procedure of construction the estimator of conditionality number can be added somehow “from the outside”. In the scheme of these methods, there are no components of finding or assessment of conditionality $M_A = \|A\| \times \|A^{-1}\|$ (since the inverse matrix is unknown) and in itself is a computationally time-consuming procedure. It is a specific task to “embed” organically such procedure into the algorithm. The problem of evaluation of the conditionality number became especially acute during accumulation of considerable experience of solving practical problems. The existence of estimation of conditionality during the computation is an important component, because it “gives the signal” (points out) of correctness of calculations. As an additional measure in solving the conditionality problem (conditionality improvement), it can be represented in the form of directed conversion of the original problem (preconditioning), for example, by multiplication of SLAE on the left and on the right by special matrices [5, 6].

To check and control the properties of the solution algorithms (testing this class of tasks), a series of algorithms, programs and modeling problems with ill-conditioned restriction matrices, such as BLAS (Basic Linear Algebra Subroutines) were developed [6]. It is known that such test matrices include the Gilbert matrix. It is actively applied in this work for testing the algorithms. The research into the influence of disturbances (such as calculation errors) on the properties of a system remains the “corner stone” during modeling. This problem is explored in a series of scientific studies, specifically, [1, 2, 6, 7]. One of the areas that has been actively developing lately, is associated with the development of mathematical methods, algorithms and software for performing the basic operations, specifically, by the introduction of new data types (for actions with rational numbers), which prevents the accumulation of errors. These approaches are discussed in papers [8–11]. One of the drawbacks of this approach is an additional computational load in the algorithm, which causes a slowdown of computation and imposes restrictions on dimensionality of the solved problems.

But the mentioned approaches do not cover the problem, so it is advisable to develop the performance of calculations using the technology of long arithmetic on the model of rational elements, one of the variants of which was implemented in the form of the Longnum library in the C++ language [23]. The application of rational arithmetic for direct methods of solution of SLAE eliminates a computational error and makes it possible to concentrate on the properties of a model itself, for example, in [7].

Modern computers usually use the standard types of integers, dimensionality of whose does not exceed 64 bites. Overcoming such hardware limitations can be addressed via programs, specifically, the development of the own data type. The known examples of the implementation of such approach are the GMP [8], MPI [9], LIP [10], OpenSSL [11], and LibTomMath libraries [12]. The GMP and LIP libraries are time-consuming and not convenient enough to use; OpenSSL and LibTomMath have cryptographic purposes; MPI library is not currently developing [9]. It is important to develop a library that does not depend on outside developments, in addition to the standard C++ libraries and can flexibly change and adapt to specific research.

Of course, the above-mentioned methods for intensification of calculation procedures have their strengths and weaknesses. They are often used separately. It seems expedient to develop the original calculation technology, which would rationally use the strengths of these approaches.

The method proposed in [14] has a series of structural properties, specifically, the capability to analyze and address the problems of linear programming (PLP) along with the SLAE, it is applied to weakly non-linear problems, has in its elements the components for calculation and evaluation of conditionality of a system during iterations [15–18], it can be equipped with the technology of long numbers. The mentioned properties of the method and the algorithms of basic matrices in equipment with the technology of operation with long numbers gain the versatility features for using in solving a wide class of problems.

The analysis of literary sources and identification of the major problems in the organization of calculations indicate that it is possible to reach precision of performing basic operations by inclusion and successful combination of the additional procedures: preconditioning, finding evaluation and conditionality number, conducting high-precision basic operations. The efforts of researchers should be aimed at the development of the computation technology with such properties (with elements of universality).

The choice of MBM as the basic one in the development of the technology of long numbers was based on the presence of such unique property as control of calculations entering the incorrectness state, specifically, the information on direct and inverse matrix as the components of the calculation of conditionality number. Construction of an estimator of conditionality number (and pre-conditioner) and its properties (with experiments) based of MBM are covered in [18].

This indicates the feasibility of application of MBM and its algorithms as those that were tested [14–18], for "add-in" with the procedures of high-precision conducting the basic operations (technology of long arithmetic) in the study of linear systems.

3. The aim and objectives of the study

The aim of the study is to develop the technology of long arithmetic, which increases the precision of performance of the basic operations of the algorithms for studying and solution of SLAE and minimizes the module of magnitude of errors at the stage of computer modeling.

To accomplish the aim, the following tasks have been set:

– to develop an algorithm for solving the SLAE (a linear system) with the elements of analysis and control of conditionality and accumulation of errors in the course of iterations (technology of long arithmetic) when implementing the technology of long arithmetic;

– to implement the types of long integers and rational numbers with fast operations of multiplication, division, constructed on modern algorithms and to accelerate the operation of multiplication and division of operations with long numbers.

4. Technologies of long arithmetic of acceleration of performing major operations based on the algorithm for the method of basic matrices (MBM)

First of all, we would like to stress that the method of basic matrices (first mentioned in publications at the end of 1980s) was developed in the article by the equipment with the technology of long arithmetic [14]. It is possible to find the detailed substantiation of the method, its properties, the results of the computational experiments, comparisons with other methods in papers [15–17].

Let us consider the SLAE in the form

$$Au=C, \tag{1}$$

where matrix A of dimensionality $(m\times m)$, $C=(c_1,c_2,...,c_m)^{\mathrm{T}}$ is the vector column of dimensionality m, $u=(u_1,u_2,...,u_m)^{\mathrm{T}}$ is the sought-for vector of dimensionality m, T is the sign of transposition, $a_j=(a_{j1}, a_{j2}, \dots, a_{jm})$, $j=\overline{1,m}$ are the rows of matrix A. Equation (1) is supplemented with the additional SLAE of the form:

$$Iu=K, \tag{2}$$

where I is the unity-diagonal matrix of dimensionality $(m\times m)$ and $K=(1,1,...,1)^T$ is the vector of dimensionality m. It should be noted that system (2) is usually trivial, with the known properties and performs only the auxiliary role of building the initial values of the MBM elements, specifically, of the inverse matrix and solution.

That is the construction of the algorithm for solving the SLAE is based on the method of basic matrices, since according to [14], it implies the ability:

– to find the magnitude of the rank of the matrix of the system restriction (1);

– to find the solution to SLAE (1);

– to control the conditionality of a system;

– to analyze the influence of changes in the model (1) as a result of specifications (without re-calculations);

– to equip with the technology that prevents error accumulation;

– to construct initial solutions to problems based on the trivial basic matrices (2), which excludes time-consuming initial calculations;

– to apply the system of analysis to the problems that imply multi-step or multiple calculations on the models with insignificant changes.

We will remind briefly [14] that the proposed method of artificial basic matrices (MABM) is based on the idea of the ordinal basic matrix. Basic matrices during the iterations are sequentially changed by the input-output of rows-normals of problem restrictions from them.

Submatrix A_b, composed of m linearly independent restriction of row-normals ($i_1, i_2, ..., i_m$), will be called artificial base, and solution u_0 to the corresponding to them equation system

$$A_b u = C^0,$$

where

$$C^0 = (c_{i_1}, c_{i_2}, ..., c_{i_m})^{\mathrm{T}},$$

will be called artificial base.

Let us assume that: e_{ri} is the elements of matrix A_b^{-1}, inverse to A_b; $u_0=(u_{01}, u_{02}, ..., u_{0m})^{\mathrm{T}}$ is the basic solution; $\alpha_r=(\alpha_{r1}, \alpha_{r2},..., \alpha_{rm})$ is the vector of development of vector-normal of restriction $a_r u \le c_r$ by the rows of basic matrix A_b; $\Delta_r=a_r u_0-c_r$ is the mismatch of the r-th restriction (1) at the vertex. All the introduced elements in new basic matrix $\overline{A}_b$ different from A_b by one row will be designated by the overbar at the top.

According to Theorem 1 [14], the corresponding ratios were established between the coefficients of expansion of normals of constraints, elements of inverse matrices, basic solutions, mismatches between restrictions in two adjacent basic matrices.

Based of them, the scheme of determining the rank of the system (1) and the solution of the system of equations, successive changes of basic matrices and corresponding artificial solutions are constructed.

The preliminary results of application of the technology of long arithmetic in conjunction with the method of basic matrices for analysis of the SLAE properties (with implementation) were examined in [22]. "Long arithmetic" of the representation of rational numbers when performing basic operations eliminates the accumulation of errors, which in combination with the conditionality control in the course of computing is effective, especially in the study of ill-conditioned systems. For the latter, even small errors in the representation of the model and the calculation have an essential impact.

In the subsequent versions, the technology repeatedly underwent improvements and both programmatically and algorithmically, the relevant calculation experiments were conducted.

The subject of research will be the properties of the new algorithm and the computer implementation of the method of artificial basic matrices [14] for model (1) in the Visual C++ environment using rational arithmetic, in which multiplication of long numbers is made with the help of the Strassen method based on the fast algorithm of discrete Fourier transform [19].

The concept of representation of integers. Modern computers usually use the standard types of integers, the size of which does not exceed 64 bytes. Overcoming such hardware limitations can be addressed programmatically, specifically, by the development of its own data type [22]. In the Longnum library developed in the language C++, the types of long integers longint3 and corresponding rational numbers and longrat3 with fast operations of multiplication, division, constructed on modern algorithms, were implemented.

For the realization of long integers of arbitrary size and appropriate rational numbers in C++, the object-oriented approach (OOA) was used [21]. It involves the following: the real object of the domain is placed in line with the so-called *object type* or *the class* of objects, which is a generalization of a structural type.

Let us use the definitions [21].

Definition 1. Class in C++ is a programming structure consisting of data (data elements or fields) and subroutines that operate on these fields and describe the properties of the corresponding object of the modeled subject area.

The general concept of long or multiple precision arithmetic, the notion of a long integer is described in [12]. But for clarity, we will introduce the concepts:

A long integer or an integer of arbitrary size will be called an integer not limited to ranges of standard computer types. For example, in respect to the language C++, such numbers are fundamentally impossible to store in a variable of type _ _int64 or unsigned_int64 that are limited to ranges from -2^{63} to 2^{63} and from 0 to 2^{64}-1, respectively.

Long or multiple precision arithmetic is the arithmetic of long integers.

An integer of an arbitrary size is programmatically represented as class longint3. Rational number as a pair (numerator of longint3 class, denominator of longint3 class) is programmed in the form of longrat3 class. An integer is stored in a dynamic array, the maximum length of which is 4096. The elements of the vector are 16-byte unsigned numbers of the standard C++ *unsigned__int16* type and are, in fact, the "numbers" in the number system with base $BASE=2^{16}$. Thus, for example, an integer from the range from 2^{32} to 2^{64}-1 will be stored in an array of size 3.

Arrays of "numbers", the number sign and operations on numbers of such structure are organized in the form of longint3 class. Specifically, in the longint3 class, the following operations with integers are implemented: addition, subtraction, multiplication and division with residue, comparison, conversion into the character string of the char type. During the development of the Longnum library [22], the elements of the STL (Standard Template Library) were used [13].

Declaration of the longint3 class

```
class longint3
{
public:
unsigned __int16 *s;
unsigned int l;
longint3();
longint3(unsigned n);
longint3(unsigned __int16 *p,int n);
longint3(const char *s1);
longint3(const char *s1,int n);
```

```
longint3(const longint3 &a);
longint3 operator=(const longint3 &a);
~longint3();
char* text(void);
friend longint3 operator+(longint3 &a,longint3 &b);
friend longint3 operator-(longint3 &a,longint3 &b);
friend longint3 operator*(longint3 &a,longint3 &b);
friend longint3 operator/(longint3 &a,longint3 &b);
friend ostream& operator<<(ostream &os,const longint3 &a);
friend istream& operator>>(istream &is,longint &a);
friend longint3 karatsuba2(longint3 &u1,longint3 &u2,unsigned int n);
friend longint3 tomakuka(longint3 &u1,longint3 &u2);
friend longint3 fastmul2(longint3 &A, longint3 &B);
friend __int8 divmod(longint3 &A, longint3 &B, longint3 &Q, longint3 &R) ;
void mult2(int n);
void mult(int m);
void div(int m);
friend int longcmp2(longint3 &a,longint3 &b);
unsigned int toint();
void print();
void operator+=(longint3 &b);
void operator-=(longint3 &b);
void operator*=(longint3 &b);
friend void savebint2(FILE *f,longint3& a);
friend void loadbint2(FILE *f,longint3& a);
friend longint3 ncd(longint3 a,longint3 b);
};
```

Representation of rational numbers. A rational number is represented as a pair: the numerator and denominator of the longint3 type and is organized in the form of the longrat3 class. This class is also equipped with the operations of addition, subtraction, multiplication, division, reducing, comparison, representation as a decimal number with specified precision, conversion in the type double, conversion into the character string of the char type. Operations '+', '-', '*', '/' are constructed using the appropriate operations on numbers of the longint3 type. These classes were compiled as a dynamic library and tested for precise solution of the systems of linear algebraic equations.

Declaration of the longrat3 class

```
class longrat3
{
public:
    __int8 sign;
    longint3 num;
    longint3 denom;
    longrat3():num("0"),denom("1"){sign=0;}
    longrat3(char *s);
    longrat3(double d);
    longrat3(char *s1,char *s2);
    longrat3(const longrat3 &l1):num(l1.num),denom(l1.denom),sign(l1.sign){}
    longrat3& operator=(const longrat3 &a);
    friend ostream& operator<<(ostream &f,const longrat3 &a);
    void text(char *s);//return s as 'p/q';
    friend void savebrat3(FILE *f,longrat3& a);
    friend void loadbrat3(FILE *f,longrat3& a);
    friend longrat3 operator+(longrat3 &a,longrat3 &b);
    friend longrat3 operator-(longrat3 &a,longrat3 &b);
    friend longrat3 operator*(longrat3 &a,longrat3 &b);
    friend longrat3 operator/(longrat3 &a,longrat3 &b);
    void operator+=(longrat3 &a);
    void operator-=(longrat3 &a);
    void operator*=(longrat3 &a);
    void operator/=(longrat3 &a);
    char* decimal(unsigned __int32 size);
    operator double();
    friend __int8 longrabscmp(longrat3 &a,longrat3 &b);
    ~longrat3(){};
    void reduce(void);
};
```

Operations '+' and '-' are implemented by classic algorithms of addition and subtraction; multiplication and division and significantly improved in comparison with the traditional algorithms for multiplication and division in a column and optimized both for time and for the use of computer resources.

Acceleration of operations of multiplication and division of long numbers.

Arithmetic operations with precise rational numbers are based on multiplication of long integers. To reduce rational fractions, division of a long integer by a long integer was implemented.

Multiplication of long integers using the Strassen method based on the discrete Fourier transform.

The product of integers

$$A=a_0+a_1BASE+...+a_{n-1}BASE^{n-1}$$

and

$$B=b_0+b_1BASE+...+b_{m-1}BASE^{m-1}$$

in a number system with $BASE$ can be interpreted as product of polynomials

$$(a_0+a_1x+...+a_{n-1}x^{n-1})(b_0+b_1x+...+b_{m-1}x^{m-1})=$$
$$=c_0+c_1x+...+c_{n+m-1}x^{n+m-1},$$

where c_i is the components of the vector – convolutions of vectors $(a_0, a_1, ..., a_{n-1})$ and $(b_0, b_1, ..., b_{m-1})$.

Let us use the definitions [18].

Definition 2. A cyclic convolution of vectors a and b is called vector $c=a\times b$ with coordinates

$$c_i = \sum_{k+l\equiv i(\mathrm{mod}\, m)} a_k b_l.$$

Definition 3. Discrete Fourier transform (DFT) of vector $(a_0, a_1,..., a_{N-1})$ is determined as a complex vector with coordinates $(y_0, y_1, ..., y_{N-1})$:

$$y_k = \sum_{j=0}^{N-1} a_j \omega^{kj},$$

where ω is the main complex root of the N-th power of unity,

$$\omega = \cos\frac{2\pi}{N} + i\sin\frac{2\pi}{N}.$$

Comment: The inverse discrete Fourier transform (DFT^{-1}) can be calculated from formula:

$$a_k = \frac{1}{N}\sum_{j=0}^{N-1} y_j \omega^{-kj}.$$

The Strassen method [20] of multiplication of long numbers is based on.

*Theorem of convolution [20]***.** Fourier transform from a cyclic convolution of two vectors is a scalar product of the Fourier-images of these vectors:

$$c = a \times b \iff F(c) = F(a) * F(b),$$

And then

$$c = F^{-1}(F(a)*F(b)).$$

Here, * means component-wise multiplication of vectors.
Thus, to multiply long integers A and B, it is enough:
1. To calculate coefficients of convolution

$$c_i,\ i = \overline{0, n+m-1}.$$

2. To make all transfers, for all coefficients to be smaller than *BASE*.

The theorem of convolution allows constructing the effective algorithm for calculation of convolution coefficients at step 1 with the help of DFT:
1. 1. Calculate $F(a)$ and $F(b)$.
1. 2. To multiply the resulting vectors component-wise.
1. 3. To calculate inverse DFT from scalar product.

Multiplication of polynomials is reduced to scalar product of corresponding vectors. It is assumed that at every step the sizes of the vectors are the same and equal to N. The complexity of the multiplication algorithm has the order of $O(N\log N)$. Moreover, the optimal performance is reached only by the options of FFT (fast algorithm of discrete Fourier transform) that operate on vectors of the size of $N=2^k$, so the vectors in these situations must be supplemented by zeros. The Strassen method is implemented in the longint3 class of the Longnum library.

Division of long integers. The operation of division of long numbers is used in the operation of reduction of rational numbers. The "school" method of division "in column", the complexity of which is O(nm), where n and m are the number of digits of a dividend and a divisor, respectively, is taken as a base.

Here are the main stages of the algorithmic scheme for finding the magnitude of rank, the initial basic matrix and solution of the non-degenerated system (1) with the rational elements, based on the technology of precise calculations. The algorithm can be applied in the conversion of an inverse matrix during the iteration process. It is advisable to use such conversion at the accumulation of significant errors when finding the elements of the method. Based on the results of the given algorithm, it is possible to construct the analytical representation of general solution of the corresponding system of linear algebraic inequalities (SLAE) [14].

Algorithms MABM

At the input: matrix of coefficients of A of dimensionality $m\times m$, vector of right parts of SLAE C of dimensionality m and unity basic matrix I of dimensionality $m\times m$, representing dynamically created arrays of the elements of the longrat3 type. The following steps will be subsequently performed with using the operations of comparison, addition, subtraction, multiplication and division of the longrat3 class.

Step 1. We perform simplex iterations on substitution of the rows of basic matrix I of system (2) with the normals of restrictions of system (1), according to the ratios of theorem 1 [14].

We find the corresponding elements of the method: vectors of expansion by the rows of the basic restriction matrices (2), inverse basic matrix, artificial basic solutions $u_0^{(k)}$, where k is the number of iterations.

Step 2. Check the number of iterations r of the substitution of the rows of the additional system with the rows of the main system, which meet the conditions of non-degeneration, i. e. $\alpha_{lk}^{(i)} \neq 0$ number determines the rank of the main system, (Consequence 2–3 of theorem 1 [14]).

Step 3. If the number of iterations, for which $\alpha_{lk}^{(i)} \neq 0,$ is equal to m, proceed to the next step. Otherwise, to the last but one step.

Step 4. Find the only solution (consequence 1 of theorem 1 [14]). According to the ratio: $A_б^{-1}c^0 = u^0$.

Step 5. Meeting condition $r<m$ means violation of condition of a single solution by the scheme of the method, i. e., the models requires refinement and further analysis of solvability (consequence 1 of theorem 1 [14]).

Last step. Generation of original information according to the results of analysis (1) in the form of an array of elements of the longrat3 type.

The algorithm can be applied in re-calculation of the inverse matrix during the iteration process of finding an optimal solution. It is advisable to perform such conversion at the accumulation of significant errors when finding the elements of the method. Based on the results of the presented algorithm it is possible to construct the analytical representation of the general solution of SLAE.

5. Results of research into computational properties of software implementation of the methods and algorithms

We explored two SLAE with ill-conditioned matrices, specifically, the Gilbert matrix A_1 with elements

$$a_{ij}^{(1)} = 1/(i+j-1), \quad i = \overline{1,m}; \quad j = \overline{1,m}$$

and matrix A_2 with elements

$$a_{ij}^{(2)} = \begin{cases} m-i+1, & \text{at } i > j, \\ m-i, & \text{at } i = j, \\ m-j+1, & \text{at } i < j. \end{cases}$$

Matrix A_1 was explored with such variations of vector c_1 of the right parts:
1) $c_i^{(1)} = 1,\ i = \overline{1,m};$
2) $c_i^{(1)} = (-1)^{i-1},\ i = \overline{1,m};$
3) $c_i^{(1)} = (1+(-1)^{i-1})/2,\ i = \overline{1,m}.$

Vector of the right parts c_2 for matrix A_2 is determined from ratio:

$$c_i^{(2)} = \begin{cases} m-i, & \text{at } i \leq 2, \\ m-i+1, & \text{at } i > 2. \end{cases}$$

Solution of SLAE using the Longnum library. In the process of solving the SLAE of dimensionality of $m=50$ with the Gilbert matrix and unity right part, using precise computations, we obtained the precise solution (Table 1).

Table 1

Precise solution of SLAE by the method of basic matrices with Gilbert matrix, m=50

Components of solution in precise form	Components of solution with floating point
-50	-50
124950	124950
-77968800	-7.79688e+007
21580031200	2.158e+010
-3350299843800	-3.3503e+012
331679684536200	3.3168e+014
-22701631741588800	-2.27016e+016
1135544885686411200	1.13554e+018
-43221677211439026300	-4.32217e+019
1290780705857666723700	1.29078e+021
-30978736940584001368800	-3.09787e+022
609077811418589580631200	6.09078e+023
-9965189747931923971993800	-9.96519e+024
137448859777688253128506200	1.37449e+026
-1615725372080580281673868800	-1.61573e+027
16336778762148089514702451200	1.63368e+028
-143202076336954347152313673800	-1.43202e+029
1095570210314899866968046826200	1.09557e+030
-7357903634707475649760709548800	-7.3579e+030
43597107685981413891518442451200	4.35971e+031
-228884815351402422930471822868800	-2.28885e+032
1068648151493282514317100869131200	1.06865e+033
-4451228664071193282775362297868800	-4.45123e+033
16584823623599852477032588070131200	1.65848e+034
-55397917798274507232310242095368800	-5.53979e+034
166193753394823521696930726286106400	1.66194e+035
-448428115668872934282842669742393600	-4.48428e+035
1089391211041939597551322864353606400	1.08939e+036
-2384432803760163710966926065345393600	-2.38443e+036
4703655197905007843631546185978606400	4.70366e+036
-8362053685164458388678304330628633600	-8.36205e+036
13391467868333092050131020150715366400	1.33915e+037
-19302545482089495962884165764117071100	-1.93025e+037
25010001538317978699384350682432678900	2.501e+037
-29077372030708791844266926744973633600	-2.90774e+037
30264203542166293552196189061095006400	3.02642e+037
-28115818722814982590157570701819743600	-2.81158e+037
23227896987219682475871594202891256400	2.32279e+037
-16986606106997219317534905455853993600	-1.69866e+037
10933522273997552736270001605050006400	1.09335e+037
-6150106279123623414151875902840628600	-6.15011e+036
2996393243665822472451151912210871400	2.99639e+036
-1250195820486420260614539573348753600	-1.2502e+036
440171703156657430859959579367246400	4.40172e+035
-128231839142745243287715497295003600	-1.28232e+035
30079073379162464474896227760556400	3.00791e+034
-5458584204914171246862075359193600	-5.45858e+033
719080128397475705222663616806400	7.1908e+032
-61171747033813037423455759068600	-6.11717e+031
2522283613639104833370312431400	2.52228e+030

The SLAE were solved using the Gaussian methods and the MABM with the Gilbert matrix of dimensionality of m=100, the reference to precise solution was published in [22].

As a result of the experiments for SLAE with the Gilbert matrix A_1 of dimensionality of m=100 and the unity right part c_1, we obtained precise solutions (Table 2) both by the Gaussian method, and by the method of the coinciding artificial basic matrices. Tables 2–4 show the time of performance of the procedure of precise solution computation using the computer with the processor AMD Athlon(tm) 64X2 Dual Core Processor 4200+, 2.21 GHz, 3 GB OSD.

For SLAE with the matrix of coefficients A_2, calculations with the use of the Gaussian method and the MABM were carried out. The results of the computational experiment are shown in Table 3.

Table 2

Solution of SLAE with the Gilbert matrix A_1 in exact numbers

Restriction vector	Dimensionality, m	Gaussian method with selection of maximum element		Method of artificial basic matrices	
		Minimal leading element	Performance time, sec	Minimal leading element	Performance time, sec
$(1,1,\dots1)^T$	50	1.79177e-051	28.5	5.56135311E-59	31.6
$(1,0,\dots1,0)^T$	50	6.7767e-054	29.3	5.56135311E-59	32.1
$(1,-1,\dots1,-1)^T$	50	6.7767e-054	29.2	0.556135311E-58	31.8
$(1,1,\dots1)^T$	60	1.66524e-065	63.3	0.1416498617E-70	68.3
$(1,1,\dots1)^T$	100	6.35121e-109	666.2	0.708461361E-119	616.2

Table 3

Solution of SLAE with matrix A_2 in exact numbers

Dimensionality m	Gaussian method with selection of maximum element		Method of artificial basic matrices	
	Leading element	Performance time, sec	Leading element	Performance time, sec
50	9.03917e-059	4.79	0.2040816326E-1	3.99
60	1.06745e+076	9.74	0.090909090E-1	6.52
61	6.18569e+077	9.41	0.030303030E-1	7.13
100	2.47997e+151	63.9	0.1010101010E-1	31.99
101	2.42961e+153	66.2	0.1E-1	31.79

As a result of the experiments for SLAE with the Gilbert matrix of dimensionality m=100 and unity right part c_1 (using the Strassen method), we obtained precise solutions by both the Gaussian method, and by the method of artificial basic matrices that are consistent with [2].

Table 4

Value of minimal leading element and time of performance of experiments for MABM with SLAE of varying dimensionality

Restriction method	Dimensionality, m	Minimal leading element	Performance time, sec	
			Previous version of Longnum	New version of Longnum
$(1,1,\dots1)^T$	50	5.56135311E-59	494.36	31.6
$(1,1,\dots1)^T$	60	0.1416498617E-70	538.92	68.3
$(1,1,\dots1)^T$	100	0.708461361E-119	1498.48	616.2
$(1,1,\dots1)^T$	120	0.8034028680E-143	3090.72	1397.4

Thus, Table 4 shows that for the MABM with the SLAE with the Gilbert matrix, the minimal leading element decreases at an increase in dimensionality, which is associated with the property of ill-conditionality of the Gilbert matrix. In addition, the time of performance of calculations increases, which is caused by both an increase in the number of arithmetic operations on rational numbers and an increase in mantissas of numerators and denominators of rational numbers. It should be noted that according to the data of comparison of Table 4, the time of computation experiments is significantly shorter for the new version of the Longnum library, which is an improvement of the longint3 class.

6. Discussion of results of the computational experiment

Application of a new class of longint3 makes it possible (Table 4):

1) to accelerate precise computations by 2.5 times: the SLAE with the Gilbert matrix of dimensionality of 100 was solved with the MBM in exact numbers in 1397.4 sec, whereas the time of performance with the old class of longint2 on the same computer was 3731.94 seconds [22];

2) to solve ill-conditioned SLAE of large dimensionality.

Compared to the old version of the Longnum library of precise computations, the following was done in the new version:

– implementations of arithmetic operations with long integers were optimized taking onto account the new standard programming language C++17;

– outdated functions (itoa, strcpy, sprintf, strcat) of standard C++ libraries were replaced by the new, more effective and safe analogues.

Thus, the advantage of using the new version of Longnum compared with the previous version [22] is the higher computational effectiveness.

The disadvantage of using precise calculations, specifically, the Longnum library, when solving SLAE is that the computation volume greatly depends not only on dimensionality of the SLAE, but also on the properties (conditionality) of the matrix of SLAE itself (Tables 2–4).

The MABM in combination with the precise computation technology of the Longnum library can be particularly useful in scientific research into the properties of SLAE and SLAN, mathematical modeling, and applied research in science and technology, which relate to the necessity of solution of ill-conditioned SLAE.

The presented study is the development of approaches to the organization and performance of the basic operations (presented in papers [8, 14]. Specifically, the technology of high-precision computation with rational numbers was combined with the control of conditionality of the system according to the scheme of the MBM algorithm, and software implementation (data type organization) was improved.

7. Conclusions

The results of the computational experiment on realization of the algorithms for conducting high-precision computations (the technology of long arithmetic) using the method of artificial basic matrices provide grounds to argue that:

1. The algorithm of SLAE (linear system), which effectively applies the value of the leading element of the method and elements of inverse matrix in order to carry out control of conditionality of the system (exceptional property of the MBM elements) was developed and implemented.

2. We implemented: the types of long integers longint3 and corresponding rational numbers and longrat3 with fast operations of multiplication, division, constructed on modern algorithms, which prevents the accumulation of errors in the course of iterations (additional equipment of the MBM algorithm).

The representation (for making computations) of rational numbers as a pair: numerator and denominator of the longint3 type was developed and organized in the form of the longrat3 class (improved compared with the previous version of the algorithm); the acceleration of the operations of multiplication and division of long numbers (for the new proposed version of the Longnum library in comparison with the previous version) was achieved. It should be noted that the additional capabilities of control of computations and measures to increase the precision of operations of the long arithmetic technology, in general, caused a slight slowdown of the algorithm operation and imposed restrictions on dimensionality of the solved SLAE.

It was established that the developed algorithms can serve as testing programs to verify the precision of carrying out computations using other algorithms for the problems of medium dimensionality.

References

1. Kahaner D., Mouler K., Nesh S. Chislennye metody i programmnoe obespechenie. Moscow: Mir, 2001. 575 p.
2. Demmel' Dzh. Vychislitel'naya lineynaya algebra. Teoriya i prilozhenie. Moscow: Mir, 2001. 430 p.
3. Han D., Zhang J. A comparison of two algorithms for predicting the condition number // Sixth International Conference on Machine Learning and Applications (ICMLA 2007). 2007. doi: https://doi.org/10.1109/icmla.2007.8
4. Ebrahimian R., Baldick R. State Estimator Condition Number Analysis // IEEE Power Engineering Review. 2001. Vol. 21, Issue 5. P. 64–64. doi: https://doi.org/10.1109/mper.2001.4311389
5. Nishi T., Rump S., Oishi S. A consideration on the condition number of extremely ill-conditioned matrices // 2013 European Conference on Circuit Theory and Design (ECCTD). 2013. doi: https://doi.org/10.1109/ecctd.2013.6662260
6. BLAS (Basic Linear Algebra Subprograms). URL: http://www.netlib.org/blas/sblat1
7. Li H., Yang H., Shao H. A note on the perturbation analysis for the generalized Cholesky factorization // Applied Mathematics and Computation. 2010. Vol. 215, Issue 11. P. 4022–4027. doi: https://doi.org/10.1016/j.amc.2009.12.009
8. The GNU Multiple Precision Arithmetic Library. URL: https://gmplib.org/
9. Multiple Precision Integer Library (MPI). URL: https://github.com/servo/nss/tree/master/lib/freebl/mpi
10. OpenSSL Cryptographic Toolkit. URL: http://openssl.org
11. Large Integer Package. URL: https://github.com/luckyaibin/BigInt/tree/master/freelip
12. Denis T., Rose G. BigNum Math. Implementing Cryptographic Multiple Precision Arithmetic. Syngress, 2006. 291 p. doi: https://doi.org/10.1016/b978-1-59749-112-9.x5000-x

13. Galovic Ya. C++17 STL. Standartnaya biblioteka shablonov. Sankt-Peterburg: Piter, 2018. 432 p.
14. Analiz svoystv lineynoy sistemy metodom iskusstvennyh bazisnih matric / Kudin V. I., Lyashko S. I., Hritonenko N. M., Yacenko Yu. P. // Kibernetika i sistemniy analiz. 2007. Issue 4. P. 119–127.
15. Bohaienko V. O., Kudin V. I., Skopetskyj V. V. Analysis of computational schemes for basic matrix method // Komp'yuternaya matematika. 2009. Issue 2. P. 3–13.
16. Bogaenko V. A., Kudin V. I., Skopeckiy V. V. Analiz vychislitel'nyh skhem modelirovaniya processov geogidrodinamiki // Probl. upr. i informatiki. 2009. Issue 4. P. 62–72.
17. Bogaenko V. A., Kudin V. I., Skopeckiy V. V. Ob osobennostyah organizacii vychisleniy na osnove metoda bazisnyh matric // Kibernetika i sistemniy analiz. 2012. Vol. 48, Issue 4. P. 146–155.
18. Bogainenko V., Kudin V. Building preconditioners using basis matrix method // International journal Information Content and Processing. 2014. Vol. 1, Issue 2. P. 182–187.
19. Knut D. Iskusstvo programmirovaniya. Vol. 2. 3-e izd. Moscow: Izdatel'skiy dom «Vil'yams», 2000. 788 p.
20. Krendall R., Pomerans K. Prostye chisla: Kriptograficheskie i vychislitel'nye aspekty. Moscow: URSS, 2011. 664 p.
21. Straustrup B. Yazyk programmirovaniya C++. Special'noe izdanie. Sankt-Peterburg-Moscow: «Nevskiy dialekt» - «BINOM», 2006. 1104 p.
22. Kudin V. I., Onotskyi V. V. Rozvynennia tekhnolohiyi dovhoi aryfmetyky pry pobudovi alhorytmiv doslidzhennia zadachi liniynoho prohramuvannia // Zhurnal obchysliuvalnoi ta prykladnoi matematyky. 2011. Issue 1. P. 77–84.

UDC 519.85
DOI: 10.15587/1729-4061.2019.157085

IDENTIFICATION OF THE STATE OF AN OBJECT UNDER CONDITIONS OF FUZZY INPUT DATA

S. Semenov
Doctor of Technical Sciences, Professor*
O. Sira
Doctor of Technical Sciences, Professor**
S. Gavrylenko
PhD, Associate Professor*
N. Kuchuk
PhD
Department of Theoretical and Applied Systems Engineering
V. N. Karazin Kharkiv National University
Svobody sq., 4, Kharkiv, Ukraine, 61022
E-mail: nina_kuchuk@ukr.net
*Department of Computer Science and Programming***
Department of Distributed Information Systems and Cloud Technologies*
***National Technical University «Kharkiv Polytechnic Institute»
Kyrpychova str., 2, Kharkiv, Ukraine, 61002

Проведена модернізація методів ідентифікації стану об'єктів в умовах нечітких вхідних даних, описаних своїми функціями належності. Обраний напрямок вдосконалення традиційних методів пов'язаний із принциповими особливостями вирішення цього завдання в реальних умовах малої вибірки вхідних даних. У цих умовах для розв'язання задачі ідентифікації стану доцільно перейти до менш вибагливої в інформаційному відношенні технології опису вихідних даних, заснованої на математичному апараті нечіткої математики. Цей перехід зажадав розробки нових формальних методів вирішення конкретних завдань. При цьому для багатовимірного дискримінантного аналізу розроблено методику розв'язання нечіткої системи лінійних алгебраїчних рівнянь. Для вирішення завдання кластеризації запропонована спеціальна процедура порівняння нечітких відстаней між об'єктами кластеризації і центрами групування. Обраний напрямок вдосконалення традиційного методу регресійного аналізу визначено неможливістю використання класичного методу найменших квадратів в умовах, коли всі змінні описані нечітко. Ця обставина привела до необхідності побудови спеціальної двохкрокової процедури вирішення завдання. При цьому реалізується мінімізація лінійної комбінації міри видалення шуканого рішення від модального і міри компактності функції приналежності пояснювальної змінної. Технологія нечіткого регресійного аналізу реалізована в важливому для практики випадку, коли вихідні нечіткі дані описані загальними функціями приналежності (L–R) типу. При цьому отримано аналітичний розв'язок задачі у вигляді розрахункових формул. В результаті обговорення показано, що модернізація класичних методів рішення задачі ідентифікації стану з урахуванням нечіткого характеру представлення вихідних даних дозволила проводити ідентифікацію об'єктів в реальних умовах малої вибірки нечітких вихідних даних

Ключові слова: нечіткі багатовимірний дискримінантний, кластерний, регресійний аналізи, технології зведення нечітких задач до чітких

1. Introduction

Let us state the general principles for solving the problem of identification of the state of an object. Information base is formed according to the results of measuring the values of a set of controllable parameters (features) of an object. Identification technologies provide linkage between these values and the state of an object. To solve this problem, a