

DOI:

UDC 004.4: 519.612.2

I.I. Zhulkovska, Candidate of Technical Sciences, Associate Professor, inivzh@gmail.com

O.A. Zhulkovskyi, Candidate of Technical Sciences, Associate Professor, olalz@ukr.net

E.A. Suhal, student, lamse3@gmail.com

Dniprovsk State Technical University, Kamianske

ESTIMATION OF COMPUTING ALGORITHMS QUICKLY DEVELOPMENT BY USING HIGH-PRECISION ARITHMETIC SOFTWARE

One of the ways to reduce the computational error of computer simulation results is the use of software tools implemented in specialized libraries of high-precision arithmetic of modern universal high-level programming languages, which is associated with a significant increase in the duration of the computational experiment. This paper is devoted to the study of the degree of influence of a given computational accuracy on the speed of the software implementation of a method for solving a SLAE of a different order and with different digit capacity of coefficients using modern means of high-precision computation of the Java language.

Keywords: computational error; high-precision computational tools; computation speed; software implementation; SLAE solution.

Одним з напрямків зниження обчислювальної похибки результатів комп'ютерного моделювання є використання програмних засобів, реалізованих в спеціалізованих бібліотеках високоточної арифметики сучасних універсальних мов програмування високого рівня, що пов'язано із значним збільшенням тривалості обчислювального експерименту. Справжня робота присвячена дослідженню ступеня впливу заданої обчислювальної точності на швидкість програмної реалізації методу рішення СЛАР різного порядку і з різною розрядністю коефіцієнтів при використанні сучасних засобів високоточних обчислень мови Java.

Ключові слова: обчислювальна похибка; високоточні засоби обчислень; швидкість обчислень; програмна реалізація; чисельне рішення СЛАР.

Formulation of the problem

The modern development of computer technology and computer modeling practically does not leave room for expensive and difficult to carry out physical experiments. That is why a computational experiment has become the most widespread and relevant method of studying all kinds of objects and processes in engineering and technology.

The solution of modern problems, formalized in the form of mathematical models, requires resource-intensive calculations over huge data arrays, the processing of which involves a colossal (sometimes even unacceptable for a particular computing architecture) number of iterations. In calculations with limited bit depth (and most modern computing systems are built on the basis of CPUs that support 64-bit machine arithmetic of the IEEE-754 standard [1] with 53 digits of the mantis), a rounding error can occur at each calculation step. At the same time, the accuracy of machine calculations becomes unsatisfactory, and the computational error becomes a determining factor in obtaining adequate results of computer modeling.

Thus, the researchers are facing the urgent task of reducing the size of the computational error associated with the limited representation of chips in the computer, which, in turn, is associated with a significant increase in the duration of the computational experiment and a decrease in its efficiency.

Analysis of recent research and publications

One of the modern ways to improve the accuracy of calculations is the 128-bit IEEE-format, in which the mantissa field is expanded to 113 bits. However, hardware support of this format is very expensive [2].

A more common way to reduce computational error is to use software tools implemented in specialized libraries of high-precision arithmetic of modern high-level universal programming lan-

guages. So, now a lot of software packages have been developed that provide both enhanced (128- or 256-bit arithmetic) and arbitrary computational accuracy.

Currently, there is a fairly wide range of such tools, the most famous of which are the following.

1. QD [3] — an advanced precision package that supports two data formats: double-double (mantissa 106 bits or approximately 32 decimal digits) and quad-double (mantissa 212 bits or approximately 64 decimal digits); It has high-level interfaces of the C ++ and Fortran-90 languages, which ensures the conversion of existing programs with minimal change in the source code.

2. ARPREC [3] — a package of arbitrary accuracy, including arithmetic calculation procedures, as well as many algebraic and transcendental functions; supports calculations with real, integer and complex numbers; has C ++ and Fortran-90 interfaces.

3. GMP [4] — a library of arbitrary accuracy, which has an extensive set of optimized procedures to support calculations with integers, rational and real numbers; rounding modes compatible with IEEE-754 specifications are not supported; has a C language interface.

4. MPFR [5] — GMP extension that provides multiple precision calculations with the possibility of using one of four rounding modes that comply with the IEEE-754 standard; accuracy can be set separately for each variable; Normalized numbers are not supported. released under the GNU LGPL license; possesses high speed in comparison with many analogs.

5. NTL [6] — portable C ++ — a library for solving problems of number theory, including data structures and algorithms for processing integers of any length, vectors, matrices and polynomials over integers and over finite fields, as well as arbitrary precision floating-point arithmetic; NTL's virtue is its consistent interface with a wide variety of classes representing mathematical objects.

6. MPFUN2015 [7] — a package of arbitrary accuracy, which is a development of MPFUN90, having a Fortran-90 language interface; partial support for the C ++ interface is planned; supports real and complex data types; the main declared advantages of the package include streaming security, to maintain acceptable performance when operating in extremely high accuracy, algorithms based on fast Fourier transform are used; the package includes routines for calculating algebraic, transcendental and some special functions, such as gamma function, incomplete gamma function, zeta function.

However, the classical methods of long arithmetic [8], which underlie the majority of well-known packages (GMP, MPFR, ARPREC, QD, NTL, etc.), lead to fairly slow and inefficient implementations. The fundamental reason for this is the emergence of hyphenation chains, due to which multi-bit mantissa processing algorithms become computationally complex and do not parallelize. As a result, high-precision calculations are accompanied by a large investment of time and inefficient use of computing resources.

Study goal statement

In [9—13], studies of the effectiveness of modern software tools for supporting high-precision calculations with real numbers for C / C ++ languages are quite widely presented. Other, listed above, software tools for the implementation of high-precision arithmetic are practically not considered in scientific publications.

The aim of this work is to study the degree of influence of a given accuracy on the speed of computational algorithms using modern means of high-precision arithmetic of the Java language. Java is selected as the most popular and demanded (according to the current ratings like TIOBE [14] or domestic DOU.UA) programming language.

Statement of the main material

To represent real numbers in Java, standard, hardware-supported, data types are used — float (precision 23 bits, 7 decimal places) and double (precision 53 bits, 16 decimal places).

To implement arithmetic with the desired (specified) accuracy, Java introduced the specialized `BigDecimal` class, which is a descendant of the `java.lang.Number` class and represents a floating-point number of arbitrary length and accuracy. `BigDecimal` is located in the `java.math` package. Each object of this class stores two integer values - the mantissa of a real number in the form of an object of class `BigInteger` and the non-negative decimal order of a number of type `int`. `BigDecimal` contains methods that duplicate standard Java arithmetic operations. When working with `BigDecimal` values, you can specify the desired precision (i.e., the number of decimal places).

When conducting a computational experiment [15], the time of the software implementation of the classical right-sweep method [16] was studied, which is a special case of the Gauss method applied to systems of linear algebraic equations (SLAE) with a three-diagonal matrix. Such systems are often found in the numerical solution of boundary value problems for differential equations of the second order in various problems of mathematical physics, in modeling some engineering problems, and also when solving problems of spline interpolation.

In the process of research, the matrix size was varied in the range $1 \leq n \leq 2 \cdot 10^7$, and the coefficients of the equation were generated randomly (taking into account the conditions for the diagonal prevalence of the matrix). In comparative calculations, we used standard valid data types: float (precision 23 bits, 7 decimal places), double (precision 53 bits, 16 decimal places) and the BigDecimal data type with similar precision (8 and 16 decimal places).

Computational experiments were carried out using the infrastructure shown in table 1.

Table 1. Computing experiment infrastructure

Processor (CPU)	Intel Core i5-7400 (4 cores, 3-3.5 GHz), cache 6 MB
Memory (RAM)	Kingston HyperX DDR4 (8GB, 2133 MHz, 17000 MB/s) + Kingston HyperX DDR4 (8GB, 3200 MHz, 23500 MB/s)
Operating system (OS)	Microsoft Windows 7
Development Environment (IDE)	Eclipse Java 2018-12
Development Kit (JDK)	1.8

So, the study of the solution time for SLAEs (Time, s) in the above range of order (n) using standard valid data types showed (Fig. 1) a slight ($3 \leq n \leq 4 \cdot 10^7$ s) linear reduction in the time of the program implementation of the sweep method for the hardware-supported double type in comparison with the less accurate standard type float.

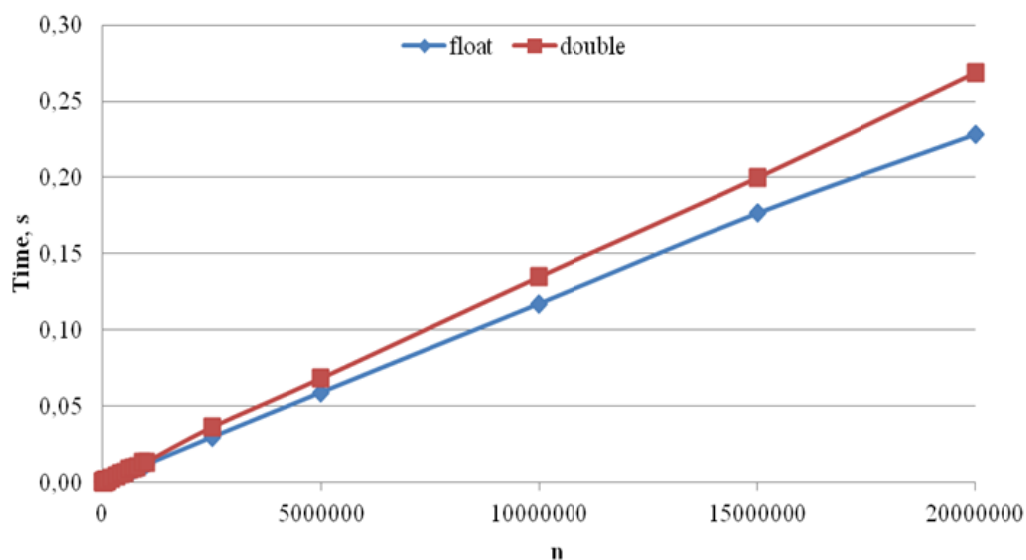


Fig. 1. Dependence of the solution time on the SLAE order for standard data types

In this case, the ratio of the calculation time when using the more accurate of these types to the calculation time for the less accurate type in the indicated range of order SLAE was 1.02–1.31 in the absence of a linear relationship.

In turn, if we compare the specialized types of Java software support BigDecimal-8 and BigDecimal-16, similar in accuracy (Fig. 2), then the software implementation of the sweep method for the more accurate of these types compared to the less accurate in the given range of order changes of SLAE was 2.3 Ч 10⁻²–3.4 Ч 10¹ s.

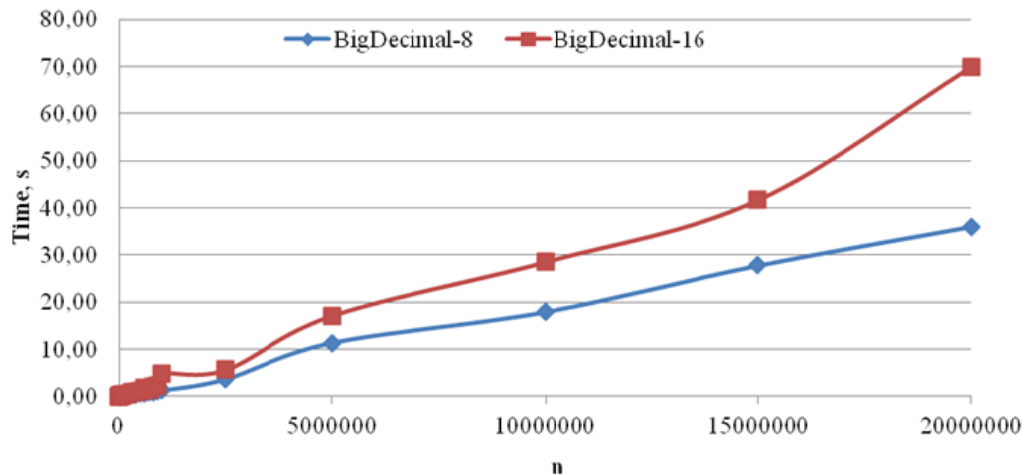


Fig. 2. The dependence of the solution time on the SLAE order for specialized data types

Moreover, the ratio of the calculation time when using the more accurate BigDecimal-16 type to the calculation time for the BigDecimal-8 type in the indicated range of the order of SLAEs was 1.5–7.3 in the absence of a linear relationship.

However, the most interesting results in this study are the results presented in Fig. 3, namely, the ratio of the speed of the standard hardware and specialized software implementation of calculations with huge (up to 2 Ч 10⁷) arrays of high-precision data.

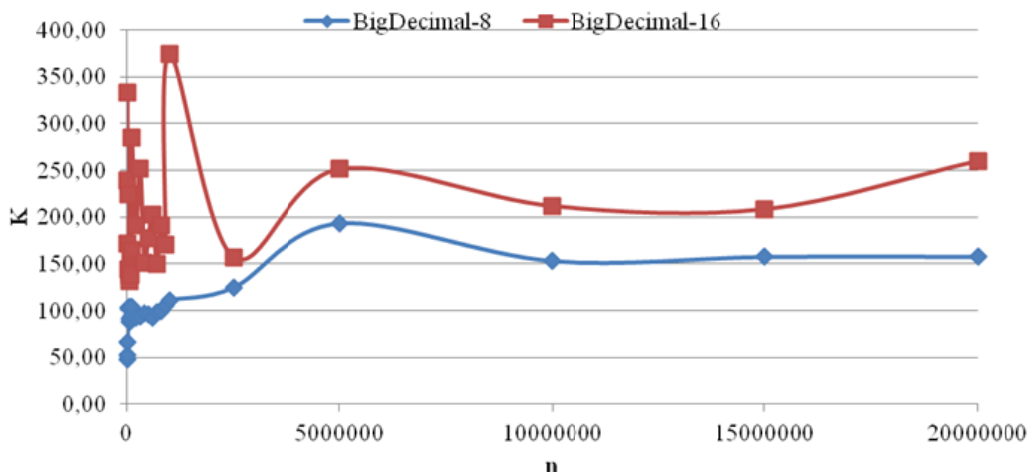


Fig. 3. Dependence of the performance drop coefficient on the order of SLAE for specialized data types

So, performance drop factors:

$$K_{BigDecimal-8} = T_{BigDecimal-8} / T_{float} ,$$

$$K_{BigDecimal-16} = T_{BigDecimal-16} / T_{double} ,$$

where, and - the SLAE solution time for the types BigDecimal-8, BigDecimal-16, float and double, respectively, significantly increase with the accuracy of the type used, i.e. there is a significant (maximum 374 times) decrease in the speed of the computational experiment. The calculated value in the studied range of changes in the order of the matrix was 48—193, and 131—374.

The present studies did not reveal a strict mathematical relationship between the decrease in the speed of the algorithm for the numerical implementation of the sweep method and the order of SLAEs, and this dependence becomes more pronounced with an increase in n from 25 Ч 105.

Conclusions and prospects for further research

The use of modern software to improve the accuracy of computer modeling is associated with a significant (several hundred times) reduction in the speed of a numerical experiment, which requires a choice between the speed of implementation of the computer model and its acceptable adequacy.

It is planned to further develop similar studies to ensure arbitrary (higher — from 16 decimal places) accuracy of the processed data, as well as the study of other hardware and software tools for implementing resource-consuming computational processes.

References

- [1] IEEE Standard for Floating-Point Arithmetic. (2008). Introduced 29 August 2008. New York, NY: Institute of Electrical and Electronics Engineers [in Russian].
- [2] Bailey D. H., Borwein J. M. High-Precision Arithmetic: Progress and Challenges [Electronic resource]. - Access Mode: <http://www.davidhbailey.com/dhbpapers/hp-arith.pdf>
- [3] Bailey D. H. High-Precision Software Directory [Electronic resource]. – Access mode: <http://www.davidhbailey.com/dhbsoftware> [in Russian].
- [4] The GNU Multiple Precision Arithmetic Library [Electronic resource]. – Access mode: <https://gmplib.org> [in Russian].
- [5] The GNU MPFR Library [Electronic resource]. – Access mode: <http://www.mpfr.org> [in Russian].
- [6] NTL: A Library for doing Number Theory [Electronic resource]. – <http://www.shoup.net/ntl> [in Russian].
- [7] MPFUN2015: A Thread-Safe Arbitrary Precision Computation Package (FullDocumentation) [Electronic resource]. – Access mode: <https://www.davidhbailey.com/dhbpapers/mpfun> [in Russian].
- [8] Brent R. Modern Computer Arithmetic / R. Brent, P. Zimmermann.(21010) – New York, NY: Cambridge University Press, 236 [in Russian].
- [9] Opanasenko, V.M., Khimich, O.M., Lisovy, O.M., Chistyakova T.V.(2011). *Razv'yazannya tasks with numbered accuracy of calculation* / Control systems and machines [in Ukrainian].
- [10] Khimich, A. , Nikolaevskaya, E, Chistyakova, T (2012) *B Programming with Multiple Precision*. / Berlin: Springer – Verlag [in Ukrainian].
- [11] Isupov, K. S. , Ivanov, A. G.(2012) *Study of the effectiveness of modern means of supporting high-precision calculations with real numbers* // "Society, Science, Innovations (NTK-2012)": Sat. materials. Section "Computing systems and software for data and knowledge." - Kirov: Vyatka State University [in Russian].
- [12] Isupov, K. S., Knyazkov, V. S. (2015) *Library of multiple precision parallel arithmetic for high-performance systems* // Supercomputer days in Russia: Tr. International Conference (2015, September 28–29 Moscow) .– CEUR Workshop Proceedings: CEUR-WS.org, – T. 1482 [in Russian].
- [13] Isupov, K., Knyazkov, V. A.(2015) *Modular-Positional Computation Technique for Multiple-Precision Floating-Point Arithmetic* // Parallel Computing Technologies, ser. LNCS. – Cham, Switzerland: Springer International Publishing. – Vol. 9251. – P. 47–61 M [in Ukrainian].
- [14] TIOBE Programming Community Index for May 2019 [Electronic resource]. – Access Mode: <https://www.tiobe.com/tiobe-index/> [in Ukrainian].

- [15] Zhulkovska, I., Suhal, E., Zhulkovskyi, (2019) O. *Analysis of the use of software approaches to reducing the computational error of the simulation results* // Zbirk tez VI All-Ukrainian. science.-practical. conf. young scientists "Information Technology – 2019". – К.: Kyiv. un-t im. B. Grinchenka, 2019. – P. 117, 118 [in Ukrainian].
- [16] Samarsky, A. A. *Theory of difference schemes*. – М.: Nauka, 1989. – 616 p [in Russian].

ОЦІНКИ ШВИДКОДІЇ ОБЧИСЛЮВАЛЬНИХ АЛГОРИТМІВ ПРИ ВИКОРИСТАННІ ПРОГРАМНИХ ЗАСОБІВ ВИСОКОТОЧНОЇ АРИФМЕТИКИ

Жульковская И.И., Жульковский О.О., Сугаль Є.О.

Реферат

Сучасний розвиток обчислювальної техніки та комп'ютерного моделювання практично не залишають місця для дорогих і важко здійснених фізичних експериментів. Саме тому обчислювальний експеримент став найбільш поширеним методом дослідження всіляких об'єктів і процесів в техніці і технологіях. В даний час перед дослідниками стає актуальною задача зниження розмірів обчислювальної похибки, пов'язаної з обмеженістю представлення чисел в ЕОМ, що, в свою чергу, пов'язано із значним збільшенням тривалості обчислювального експерименту і зниженням його ефективності.

Метою даної роботи є дослідження ступеня впливу заданої точності на швидкодію обчислювальних алгоритмів при використанні сучасних засобів високоточної арифметики мови Java.

При проведенні обчислювального експерименту досліджувався час програмної реалізації класичного методу правої прогонки, що застосовується до систем лінійних алгебраїчних рівнянь з тридіагональною матрицею.

У процесі досліджень розмір матриці змінювали в діапазоні 10^3 — $2 \cdot 10^7$, а значення коефіцієнтів рівняння генерували випадковим чином (з урахуванням умови діагонального переважання матриці). У порівняльних розрахунках використовували стандартні дійсні типи даних: float (точність 23 біта, 7 десяткових знаків), double (точність 53 біта, 16 десяткових знаків) і тип даних BigDecimal з аналогічними точностями (8 і 16 десяткових знаків).

Використання сучасних програмних засобів підвищення точності комп'ютерного моделювання пов'язане із значним (в кілька сотень разів) зниженням швидкості чисельного експерименту, що вимагає вибору між швидкодією реалізації комп'ютерної моделі і її прийнятною адекватністю. Планується подальший розвиток аналогічних досліджень для забезпечення достатньо високої (більш високою – від 16 десяткових знаків) точності оброблюваних даних, а також використання інших апаратних і програмних засобів реалізації подібних обчислювальних процесів.

Література

1. IEEE Standard for Floating-Point Arithmetic. Introduced 2008-08-29. New York, NY: Institute of Electrical and Electronics Engineers, 2008. 70 p.
2. Bailey D. H., Borwein J. M. High-Precision Arithmetic: Progress and Challenges [Електронний ресурс]. Режим доступу: <http://www.davidhbailey.com/dhbpapers/hp-arith.pdf>
3. Bailey D. H. High-Precision Software Directory [Електронний ресурс]. Режим доступу: <http://www.davidhbailey.com/dhbsoftware>.
4. The GNU Multiple Precision Arithmetic Library [Електронний ресурс]. Режим доступу: <https://gmpmath.org>.
5. The GNU MPFR Library [Електронний ресурс]. Режим доступу: <http://www.mpfr.org>.
6. NTL: A Library for doing Number Theory [Електронний ресурс]. <http://www.shoup.net/ntl>.

7. MPFUN2015: A Thread-Safe Arbitrary Precision Computation Package (FullDocumentation) [Электронный ресурс]. Режим доступа: <https://www.davidhbailey.com/dhbpapers/mpfun2015.pdf>.
8. Brent R., Zimmermann P. *Modern Computer Arithmetic*. New York, NY: Cambridge University Press, 2010. 236 p.
9. Розв'язання задач с підвищеною точністю обчислень / В. М. Опанасенко, О. М. Хіміч, О. М. Лісовий, Т. В. Чистякова // *Управляющие системы и машины*. – 2011. – № 1. – С. 9 – 18.
10. Khimich A., Nikolaevskaya E., Chistyakova T. *Programming with Multiple Precision*. Berlin: Springer–Verlag, 2012. 206 p.
11. Исупов К. С., Иванов А. Г. Исследование эффективности современных средств поддержки высокоточных вычислений с вещественными числами. «Общество, наука, инновации (НТК-2012)»: Сб. материалов. Секция «Вычислительные системы и программное обеспечение для данных и знаний». Киров: ВятГУ, 2012. 11 с.
12. Исупов К. С., Князьков В. С. Библиотека параллельной арифметики многократной точности для высокопроизводительных систем // *Суперкомпьютерные дни в России: Тр. Международной конференции (28–29 сентября 2015 г. г. Москва)*. CEUR Workshop Proceedings: CEUR-WS.org, 2015. Т. 1482. С. 110–121.
13. Isupov K., Knyazkov V. A Modular-Positional Computation Technique for Multiple-Precision Floating-Point Arithmetic. *Parallel Computing Technologies, ser. LNCS*. – Cham, Switzerland : Springer International Publishing. – 2015. Vol. 9251. P. 47–61.
14. TIobe Programming Community Index for May 2019 [Электронный ресурс]. Режим доступа: <https://www.tiobe.com/tiobe-index/>
15. Zhulkovska I., Suhal E., Zhulkovskyi O. Analysis of the use of software approaches to reducing the computational error of the simulation results // *Збірник тез VI Всеукр. наук.-практ. конф. молодих науковців «Інформаційні технології – 2019»*. Київ. ун-т ім. Б. Грінченка, 2019. С. 117–118.
16. Самарский А. А. *Теория разностных схем*. М.: Наука, 1989. 616 с.