

UDC 004.43

Yu.V. Korniyenko, Master,
M.G. Suryaninov, DEng, Prof.,
Odessa National Polytechnic University

DEVELOPMENT OF CAD IMPLEMENTING THE ALGORITHM OF BOUNDARY ELEMENTS' NUMERICAL ANALYTICAL METHOD

Introduction. Initially, the boundary elements method (BEM) developed rather slow due to the lack of software and integral equations numerical processing difficulties. The technique preferred for solid bodies' stresses and strains calculation embodies the finite elements method (FEM). Therefore, a wide spread implementation of BEM began about a decade later than the intensive work on the FEM implementation. For that instance engineers had already solved two-dimensional problems actively using the finite element method, and did fully appreciate the advantages of CAD-assisted calculations at the same time that the need to implement similar ones for three-dimensional areas. Then the researchers turned to the use of the boundary element method, allowing to one-unit reduce the problem' geometric dimension.

Analysis of recent research and publications. The first programs implementing the BEM algorithms [1...3], are locally applied and composed in FORTRAN.

That programming language is one of the first high-level languages including a translator mechanism. The user, creating a code with FORTRAN, had the opportunity to operate numbers (integers, real numbers, complex) with high accuracy as well as to use the built-in mathematical functions library. Later, some of lexical structures available to structural, functional, and sometimes even to object-oriented languages were also added to FORTRAN.

The next programs [4, 5] used more sophisticated FORTRAN versions, and later the Visual Basic [6] became the preferred tool of this kind.

Until now the numerical and analytical BEM algorithms [7] are implemented using programs written by ONPU scientists [8] in MATLAB environment, at that each program has a local character, i.e., is solving a particular problem: calculation of beams, frames, arches, etc. The matrixes' construction with these programs being done "by hand" is very time consuming.

Aim of Research is to perform a reasoned choice of programming language for new CAD system development; such a system would allow to implement algorithms for numerical and analytical boundary elements method and to create tools for visualizing both the original objects and the calculation results.

Main Body. Various BEM versions are embodied in the form of local programs written in different programming languages. The LCR2D program [2] originally written in FORTRAN, was later written over using GNU G77 FORTRAN. The BEPLANE program [4] was created in FORTRAN 77.

Originally FORTRAN standard used a fixed source text format, and beginning with FORTRAN 90 version, it is modified: free source code format, as well as the more than one variable set in each row. This allowed developers to create a code not tied to the lines positions, so its percipiency and readability are comparable to the codes in modern programming languages (C, Java).

Engineers are still using FORTRAN language for practical and scientific computations, since there exists a huge number of programs and sub-programs, collected in libraries created in this language. These programs are open to source code writing, being tested and are very useful; another ben-

DOI 10.15276/opu.1.45.2015.21

© Yu.V. Korniyenko, M.G. Suryaninov, 2015

efit is that available is a rich compendium of these programs detailed documents. FORTRAN 95 and FORTRAN 2003 versions already provide possibilities of using modern programming techniques and even object-orientation tools.

The program BEPLANE [4] creators went further with their new version BEPLANE-2002 using Visual Basic 6. Although the program's computational part is still developed in FORTRAN.

Also Visual Basic 6 is used at another CAD: BEASY [9]. This program can perform planar structures' static analysis, as well as some calculations in the field of fracture mechanics.

The Microsoft Visual Basic [6] represents a tool for software creation. For convenience, the Microsoft Corporation developed both the programming language and development environment. The Basic language, having a sufficient dialects number, is just "an ancestor" to Visual Basic language that has partly inherited the Basic's syntax and style. Also Visual Basic combines methods and objects of component- and object-oriented programming languages. The Microsoft' development environment, VB, contains tools auxiliaries to construct the user's visual interface. Binding to the Visual Basic programming environment allows developing applications only for Microsoft Windows. Practically all internal functions of Visual Basic are realized through the runtime library which performs a lot of unnecessary processes when conversion and type checking.

The new public library BEM ++ [10] is written in two high-level general purpose programming language C ++ and Python. Program code in Python is combined with already ready libraries in C ++ either is used directly as Python code.

Although the C ++ programming language is free for use, the documents on language standards (except for drafts) are not available for free.

The Python language syntax [11], in contrast to C ++, is minimal. A standard Python library has a large number of useful features. The Python Software Foundation License is free and allows using code written in any application.

Numerous projects are realized in Python, also used to implement the future applications' models. But, like most other programs written in interpreted programming languages, programs in classic Python version, are running with a relatively low speed.

The first version of CAD system developed by the authors is written in Delphi and has no means of visualizing the original calculation object. Due to the fact that the development in Delphi requires a license, there was taken a decision of moving to a higher-level language, which does not require licensing nor is tied to a specific operating system.

A programming language useful for the developer should contain a lot of options and be multifunctional. The Java is one of such multifunctional languages. This object-oriented language is created by Sun Microsystems (later acquired by Oracle) [12].

Some concepts of C, C ++, Objective C, Small Talk, Cedar (Mes) and Eiffel were used while creating the Java language. Therefore, Java and those listed languages provide possibilities to solve the same problems.

To create practical applications used by numerous users preferred is to choose the Java, representing a safe programming language with a fairly high threshold for entry.

A code written in Java language [13] is translated into byte code executed with Java machine (Java VM — JVM). The JVM is a program that, acting in an interpreter, handles byte code and transmits the control guidance to hardware. The byte code's complete freedom from both operating system and hardware represents one of the benefits of program execution in a similar way. This allows running Java-based application on a device for which a respective virtual machine is created.

Using other programming languages, the CAD realization and run in another development environment will require additional measures apart of recompiling the program text. Different graphical environments and operating systems have incompatible interfaces that creates numerous problems in user interfaces' implementation. A developer willing to build an application able to run with different platforms will face the necessity to create series of different applications and control them separately [13].

To run Java application on different platforms the developer does not need to compile every time the written code, but he should simply check and compile it into one user-friendly platform. Thus, the formed byte code will be suitable for the necessary platform operating with Java virtual processor (Fig. 1), and an application written in Java language is only once corrected and compiled.

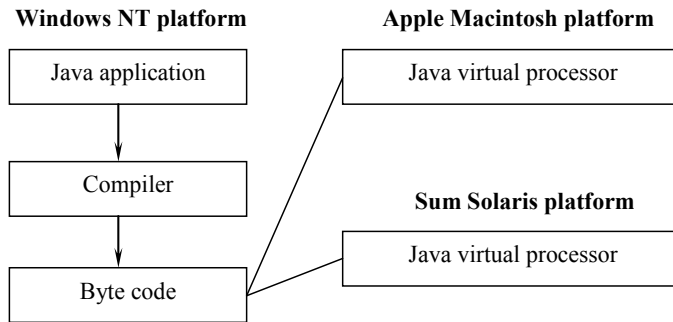


Fig. 1. Schematic representation of Java application preparation process for operation with various platforms

The Java virtual machine is completely controlling the written program use due to another peculiarity of Java language: its flexible security system. Therefore, the operations that exceed the program's authorized capacities, e.g., peer-to-peer connection either unauthorized data access command do instantly initiate the application shutdown.

Provided, most of standard algorithms (especially these related to accessible data types and similar) have been implemented earlier and are available with the programming language, the program can be developed much faster and requires less space. In such a way, one of essential Java's advantages relates to the standard data library, which volume increases with every next version so far embracing wide scope of potential allocation fields. The Java executive environment by now includes the graphics and graphic interface libraries, data bases access libraries and other typical tasks de-facto defining their use standards. The basic C++ library allows much less capacities.

Additional benefit of Java programming language, as afore said, refers to exemption of the need to buy license; in the internet (<http://java.sun.com>) everyone can freely get the Java Development Kit (JDK).

The described advantages of Java language made possible to solve efficiently the task with elaboration of a program implementing the numeric-analytical BEM algorithm thus allowing to visualize the researched objects' calculation schemes and obtained results.

Users can select the required structure type – beam, frame, arch, plate or shield (Fig. 2, a), and the calculation type — statics, dynamics, stability. Calculation task visualization example is shown with Fig. 2, b.

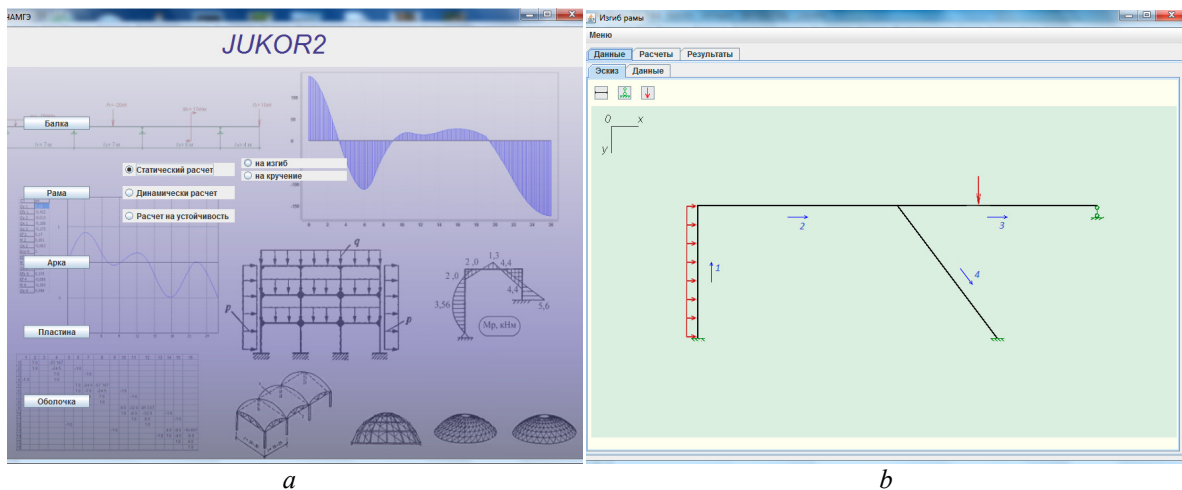


Fig. 2. Main window of a program written in Java

The program does generate the numeric-analytic BEM matrixes and resolving equations systems. All calculations are performed automatically. Several results of matrixes' building are given at Fig. 3.

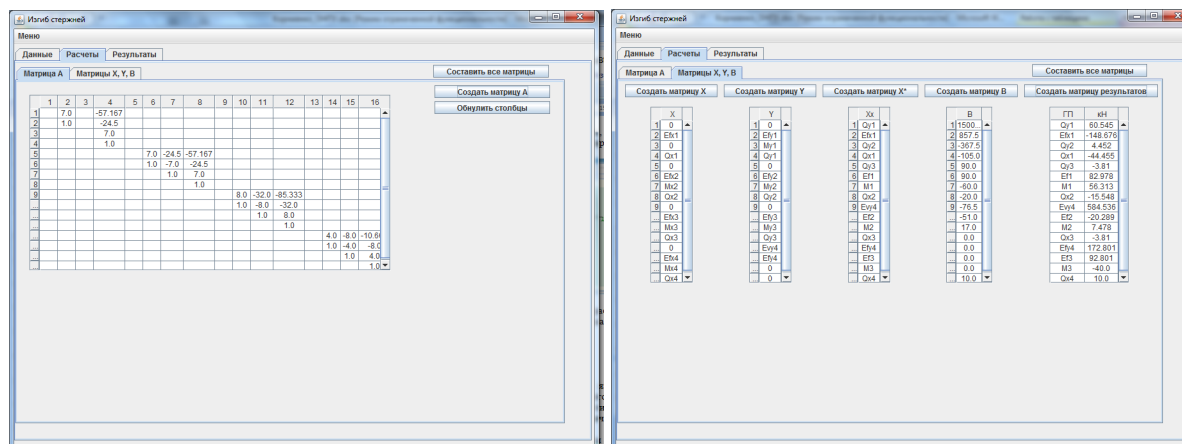


Fig. 3. Numeric-analytical BEM algorithm's matrixes generated with the program

The matrix equation's automatic solution get, the user can obtain required data for the system's selected points as well as analyze the diagrams plotted. Refer to Fig. 4 for calculation result.

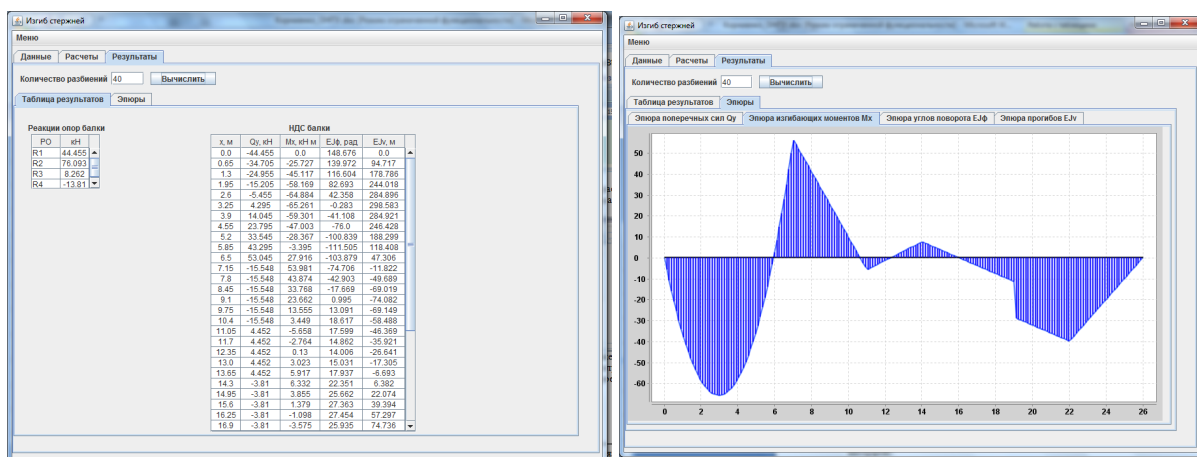


Fig. 4. Calculation results

Results. At the analytical study issue evidently shown is that the Java represents a programming language the most efficient for numeric-analytical BEM algorithm implementation for CAD needs. This language's capacities allow not only the CAD system's calculations part development but also to develop the graphic interface.

Conclusions. All afore exposed confirms that CAD systems, based onto classic BEM, were progressing along with the programming languages' development. The newest modern programs reached essential progress in results' calculation and graphic visualization, still none CAD system not yet implements the boundary elements numeric analytic method. The research makes evident that such an universal program needed for contemporary engineering purposes is worthwhile to be developed using the Java platform.

Література

1. Kerelezova, I. Numerical modeling of quasibrittle materials by means of fracture mechanics approach / I. Kerelezova // The University of Architecture, Civil Engineering and Geodesy. — Sofia, 2002.
2. Khapaev, M. LCR2D: 2D field solver for calculation of inductances, capacitances and resistances [Електронний ресурс] / М. Khapaev // Moscow State University, Faculty of Computer Science, Dep.

- of Numerical Methods. — Moscow, 2005. — Режим доступу: http://vm.cs.msu.ru/sotr/vmhap/lcr2d/lcr_man.pdf (Дата звернення: 01.09.2014).
3. Pozrikidis, C. *A Practical Guide to Boundary Element Methods with the Software Library BEMLIB* / C. Pozrikidis. — Boca Raton: Chapman & Hall/CRC, 2002. — 440 p.
 4. Абдылдаев, Э.К. Анализ методов конечных и граничных элементов на практических задачах [Электронный ресурс] / Э.К. Абдылдаев, К.С. Раматов, Батырхан Серик // Вестник КазНТУ. — 2008. — № 4. — Режим доступу: http://e-lib.kazntu.kz/sites/default/files/articles/abdyldaev_2008_4.pdf (Дата звернення: 01.09.2014).
 5. Себеста, Р.У. Основные концепции языков программирования = *Concepts of Programming Languages*: монографія / Р.У. Себеста; [Пер. с англ. Д.А. Ключина, А.В. Назаренко]. — 5-е изд. — М.: Вильямс, 2001. — 668 с.
 6. Сафронов, И.К. *Visual Basic в задачах и примерах*: монографія / И.К. Сафронов. — СПб.: БХВ-Петербург, 2008. — 394 с.
 7. Численно-аналитический метод граничных элементов / А.Ф. Дашченко, Л.В. Коломиец, В.Ф. Оробей, Н.Г. Сурьянинов. — В 2 т. — Одесса: ВМВ, 2010. — Т. 1. — 416 с. — Т. 2. — 512 с.
 8. Численные методы в механике / В.А. Баженов [и др.]. — Одесса: СТАНДАРТЬ, 2005. — 564 с.
 9. Al-Otaibi, M. The application of BEASY software to simulate cathodic protection of pipelines and storage tanks [Электронный ресурс] / M. Al-Otaibi // University of British Columbia. — 2010. — Режим доступу: <https://circle.ubc.ca/handle/2429/30264> (Дата звернення: 01.09.2014).
 10. BEM++: An open source boundary element library [Электронный ресурс] / S. Arridge et al. — Режим доступу: <http://www.bempp.org> (Дата звернення: 01.09.2014).
 11. Лутц, М. Программирование на Python: монографія / М. Лутц; [Пер. С. Маккавеева]. — 2-е изд. — СПб.: Символ-Плюс, 2002. — 1135 с.
 12. Oracle: Hardware and Software, Engineered to Work Together [Электронный ресурс] / Oracle Corporation. — Режим доступу: <http://www.oracle.com> (Дата звернення: 01.09.2014).
 13. Эккель, Б. Философия Java = *Thinking in Java* / Б. Эккель; [Пер. с англ. И. Портянкин]. — СПб.: Питер, 2001. — 876 с.

References

1. Kerelezova, I. (2002). *Numerical Modeling of Quasibrittle Materials by Means of Fracture Mechanics Approach* (PhD thesis, The University of Architecture, Civil Engineering and Geodesy). Sofia, Bulgaria: UACG.
2. Khapaev, M. (2005). LCR2D: 2D field solver for calculation of inductances, capacitances and resistances. *The faculty of Computational Mathematics and Cybernetics of Lomonosov Moscow State University*. Retrieved from http://vm.cs.msu.ru/sotr/vmhap/lcr2d/lcr_man.pdf
3. Pozrikidis, C. (2002). *A Practical Guide to Boundary Element Methods with the Software Library BEMLIB*. Boca Raton: Chapman & Hall/CRC.
4. Abdyldaev, E.K., Ramatov, K.S. and Batyrkhan Serik (2008). Analysis of methods of finite and boundary elements for practical problems. *Vestnik KazNTU*, 4. Retrieved from http://e-lib.kazntu.kz/sites/default/files/articles/abdyldaev_2008_4.pdf
5. Sebesta, R.W. (2002). *Concepts of Programming Languages* (5th Ed.). Boston: Addison-Wesley.
6. Safronov, I.K. (2008). *Visual Basic in Examples and Problems*. Saint Petersburg: BHV-Sankt-Peterburg.
7. Daschenko, A.F., Kolomiets, L.V., Orobey, V.F. and Suryaniniv, N.G. (2010). *Numerical-Analytical Boundary Elements Method*. Odessa: VMV.
8. Bazhenov, V.A., Dashchenko, A.F., Kolomiets, L.V., Orobey, V.F. and Suryaniniv, N.G. (2005). *The Numerical Methods in Mechanics*. Odessa: STANDART.
9. Al-Otaibi, M. (2010). *The Application of BEASY Software to Simulate Cathodic Protection of Pipelines and Storage Tanks* (Master's thesis, University of British Columbia). Vancouver: University of British Columbia.
10. BEM++: An open source boundary element library. Retrieved from <http://www.bempp.org>
11. Lutz, M. (2001). *Programming Python* (2nd Ed.). Beijing: Farnham: O'Reilly.
12. Oracle: *Hardware and Software, Engineered to Work Together*. Retrieved from <http://www.oracle.com>
13. Eckel, B. (2000). *Thinking in Java* (2nd Ed.). Upper Saddle River, NJ: Prentice Hall.

АНОТАЦІЯ / АННОТАЦИЯ / ABSTRACT

Ю.В. Корнієнко, М.Г. Сур'янінов. Розробка САПР, що реалізує алгоритм чисельно-аналітичного методу граничних елементів. Реалізація алгоритмів чисельно-аналітичного методу граничних елементів досі здійснювалася за програмами, написаними мовою середовища MATLAB. Кожна з цих програм носить локальний характер, тобто застосовується для вирішення конкретного завдання: розрахунку балки, рами, арки і т.д. А побудова матриць в цих програмах є трудомісткою. Метою роботи є аргументований вибір мови програмування для розробки нової САПР, що дозволяє реалізувати алгоритм чисельно-аналітичного методу граничних елементів і створити засоби візуалізації вихідних об'єктів і результатів розрахунку. Проведені дослідження показують, що з усіх основних мов програмування найбільш ефективним для розробки САПР, що реалізує алгоритм чисельно-аналітичного методу граничних елементів, є мова Java. Можливості цієї мови дозволяють розробити не лише розрахункову частину САПР, але і графічний інтерфейс для побудови геометричних моделей і інтерпретації результатів розрахунку.

Ключові слова: алгоритм, чисельно-аналітичний метод граничних елементів, мова програмування, об'єктно-орієнтована мова, код, САПР, програма.

Ю.В. Корниенко, Н.Г. Сурьянинов. Разработка САПР, реализующей алгоритм численно-аналитического метода граничных элементов. Реализация алгоритмов численно-аналитического метода граничных элементов до сих пор совершалась по программам, написанным на языке среды MATLAB. Каждая из этих программ носит локальный характер, т.е. применяется для решения конкретной задачи: расчета балки, рамы, арки и т.д. А построение матриц в этих программах является трудоемким. Целью работы является аргументированный выбор языка программирования для разработки новой САПР, позволяющей реализовать алгоритм численно-аналитического метода граничных элементов и создать средства визуализации исходных объектов и результатов расчета. Проведенные исследования показывают, что из всех основных языков программирования наиболее эффективным для разработки САПР, реализующей алгоритм численно-аналитического метода граничных элементов, является язык Java. Возможности этого языка позволяют разработать не только расчетную часть САПР, но и графический интерфейс для построения геометрических моделей и интерпретации результатов расчета.

Ключевые слова: алгоритм, численно-аналитический метод граничных элементов, язык программирования, объектно-ориентированный язык, код, САПР, программа.

Yu.V. Korniyenko, M.G. Suryaninov. Development of CAD implementing the algorithm of boundary elements' numerical analytical method. Up to recent days the algorithms for numerical-analytical boundary elements method had been implemented with programs written in MATLAB environment language. Each program had a local character, i.e. used to solve a particular problem: calculation of beam, frame, arch, etc. Constructing matrixes in these programs was carried out "manually" therefore being time-consuming. The research was purposed onto a reasoned choice of programming language for new CAD development, allows to implement algorithm of numerical analytical boundary elements method and to create visualization tools for initial objects and calculation results. Research conducted shows that among wide variety of programming languages the most efficient one for CAD development, employing the numerical analytical boundary elements method algorithm, is the Java language. This language provides tools not only for development of calculating CAD part, but also to build the graphic interface for geometrical models construction and calculated results interpretation.

Keywords: algorithm, numerical-analytical boundary elements method, programming language, object-oriented language, code, CAD, program.

Received September 8, 2014