EFFICIENT PROCESSING OF NUMERICAL ALGORITHMS THROUGH NUMERICAL ENGINEERING SOFTWARE

MILITARU, Romulus
Department of Applied Mathematics
University of Craiova
militaruromulus@yahoo.com

CĂLIN, Liviu-Adrian
Faculty of Mathematics and Computer Science
University of Craiova
adi.calin.nds@gmail.com

CĂLUGĂRU, George-Cristian
Faculty of Automation, Computer Science and Electronic
University of Craiova
calugaru.george.nds@gmail.com

GEORGESCU, Adrian
Faculty of Automation, Computer Science and Electronics
University of Craiova
georgescu.adrian.nds@gmail.com

Abstract

Numerical Engineering Software represents a new software for solving technical problems through numerical algorithms. It consists of five chapters: Matrix Algebra, Polynomial Approximations, Roots of Equations, Numerical Integration and Cauchy Problems. As of March 2000, Numerical Engineering Software has been consistently upgraded to a new more complex version. We give a survey of this new version along with the basic features that are part of the Numerical Engineering Software project.

Keywords: *Matrix Algebra, Polynomial Approximations, Roots of Equations, Numerical Integration and Cauchy Problems*

AMS Classification: 65D05, 65D30, 65F05, 65F10, 65F15, 65L05, 65-04

CR categories: D.2.2

1. **Introduction**

Numerical algorithms are an integral part of many scientific, data mining, multimedia, and high performance computing applications, being extensively studied in the area's specific literature.

Numerical Engineering Software (NES) is a relatively new software solution in the field of numerical analysis. The project members had in mind a solution that would benefit the solving of technical problems through numerical methods. The selection of the numerical algorithms, their improvements in order to achieve high accuracy and requiring a low computational cost, along with the creation of other particular methods was a long and hard process.

Numerical Engineering Software is a cross-platform application that runs on Windows, MAC OS X, Sun and Linux with proficient results. The project developers intended to create a software solution which will contain complex numerical algorithms combined with a simple, interactive interface.

The starting point of the development of our project was a desire to make numerical calculus more easy to use and understand. The next step was to start the creation of a project that will not need any prior programming knowledge and will not even require good English knowledge. The only things that you need to be acquainted with are the restrictions of the method you want to use.

The first project of such a nature combining few of the features above was called Numsoft. Numsoft was a rough version of what we had in mind. Looking at it now, it can be easily called a prototype. It had implemented a various range of numerical methods, but they were rough, having a large computational cost and covering virtually no particular cases. But, despite all these downfalls, a plan for the interactive interface we had in mind was emerging. The Numsoft project and its very limited use were implemented somewhere between March and May 2008.

The next step in the development of a numerical analysis software was the first version of Numerical Engineering Software presented at the CAMAI 2008 conference, ([10]). The mainframe for the interface was set. The possibility of the user to impose the accuracy of the results was imposed in several methods. The ".nes" files appeared because of the implementation of our own text editor. The files can be saved and processed at a later time. The particular cases were covered completely and methods were improved where necessary. The number of sections grew. The overall number of numerical methods increased. The graphic approximations section appeared and was fundamented here. The basic mathematical functions (trigonometric functions, exponential etc.) were implemented and they are used in the most familiar way possible(e.g. like in Matlab, ([6])). This version is characteristic to the July-December period 2008. Since then, NES has grown constantly and today it is safe to say that it has reached a new development level that is highly superior to that of the December 2008 version.

This new version is going to be presented in detail in this report. This one has a possibility of choosing between languages: Romanian, English and

French. The differential equations chapter was improved. The highlight of this chapter is our embedded algorithm for solving high order differential equations based on a Runge-Kutta 4-th order method.

The number of numerical methods increased further. Methods were improved and most of them benefit from the possibility of imposing the accuracy of the results. As a new feature, today we are able to represent a number with up to 15 precise decimal places.

The recent Militaru method for the calculation of the extreme eigenvalues of a real symmetric matrix was implemented. A complete visual guide was also added to this version. The complete update report of the new version will be presented in a separate section.

The new version of Numerical Engineering Software consists of the same five chapters: Matrix Algebra, Polynomial Approximations, Roots of Equations, Numerical Integration and Cauchy Problems.

The number of methods has been multiplied, precisions have been improved and the overall complexity of the program has been increased.

The program is used in the laboratories of the Faculty of Automation, Computers and Electronics from Craiova with proficient results, the project continuing to grow on a weekly basis, which makes it very dynamic.

2. **Complete list of numerical methods present in the new version of Numerical Engineering Software**

2.1 *Matrix algebra*

The branch of mathematics which deals with the study of matrices is called matrix theory. Though it was initially a subdivision of linear algebra, it has evolved in such a manor that it now covers graph theory, algebra, combinatorics and also statistics. In order to better express the role of matrices we reiterate the fact that the PageRank link analysis algorithm used by Google to assign each element on the web a certain weight and thus create the order in which the pages appear on a search is based on the evaluation of a stochastic matrix of very large dimensions. A linear system is a mathematical model of a system based on the use of a linear operator. The properties described by a linear system are much simpler than the general nonlinear case. It can be said that linear systems can describe properties that occur during an ideal state of a phenomenon. Linear systems find important applications in automatic control theory, signal processing and telecommunications ([1], [2], [5]).

The methods included in the first chapter are:

- Gauss elimination method of inverting a real square matrix;
- iterative method of inverting a real square matrix;
- iterative Seidel-Gauss method for solving large linear systems;
- iterative Seidel-Gauss method for solving sparse linear systems;
- LR factorization for solving linear systems;
- LR factorization for solving tridiagonal matrix linear systems;
- LR factorization for solving pentadiagonal matrix linear systems;

- Fadeev method for determining the characteristic polynomial of a real matrix;
- Danilevski method for obtaining eigenvalues and the eigenvectors corresponding to real eigenvalues, for a real square matrix (covering all particular cases);
- LR method for the calculation of eigenvalues and the eigenvectors corresponding to real eigenvalues of a real square matrix (covering all particular cases);
- Militaru method for the calculation of the extreme eigenvalues of a real square symmetric matrix (based on successive approximations);

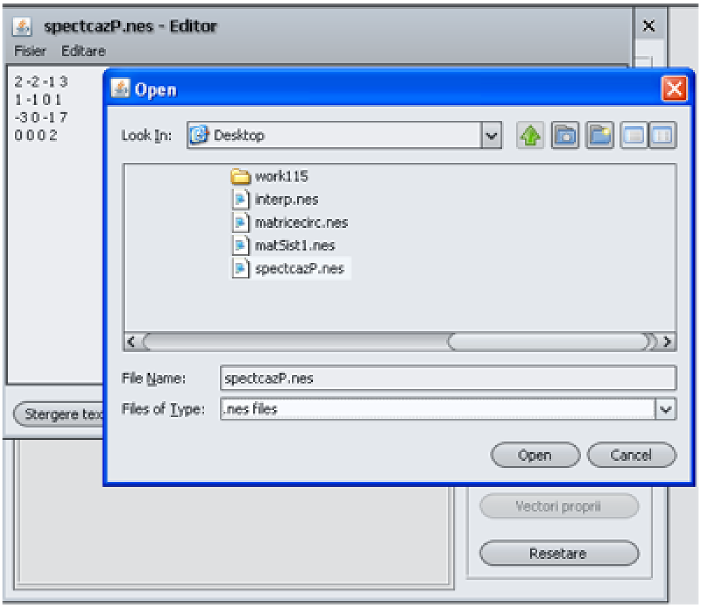

Figure 1. *Opening a matrix through a .nes file (Romanian version)*

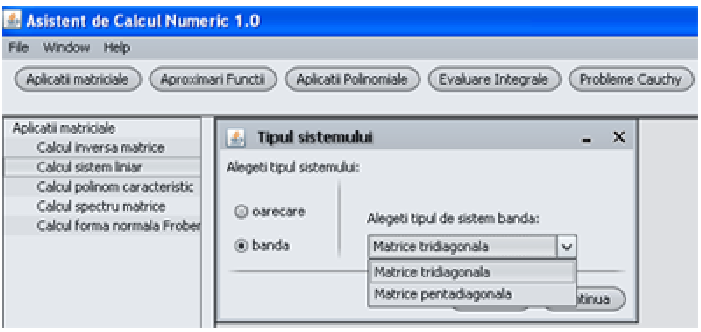

Figure 2. *Choosing of the type of linear system to be processed (Romanian version)*

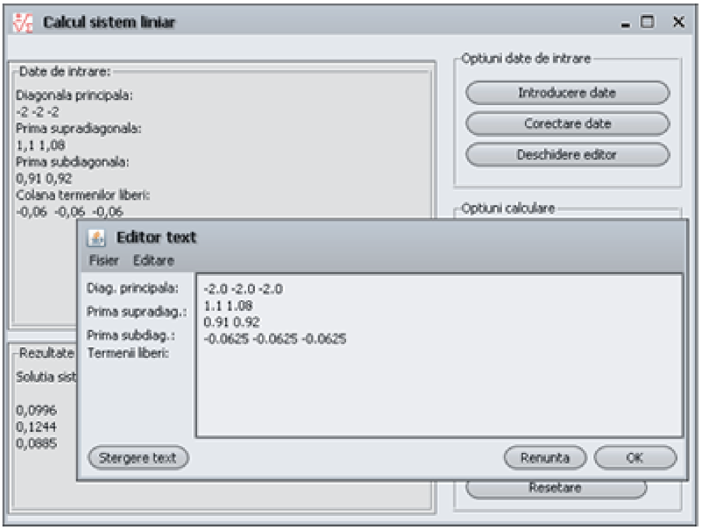

Figure 3. *The calculation of a linear tridiagonal systems (Romanian version)*

2.2 *Polynomial approximations*

The problem of interpolation is the one of finding a "nice" function of a single variable that has given values at specified points. It is important both as a theoretical tool for the derivation and analysis of other numerical algorithms (finding zeros of functions, numerical integration, solving differential equations etc.) as a means to approximate functions known only at a finite set of points. The most famous application of interpolation is the processing of data obtained through sampling ([4], [11], [15]).

The methods included in this chapter:
- Lagrange's interpolating method;
- Newton's interpolating method;
- Discrete least square approximation method;
- Cubic Spline interpolation method (free and imposed boundary).

2.3 *Locating roots of equations*

The method included in this chapter is the famous and reliable Bairstow method for the calculus of the roots (real or complex) of an algebraic equation, having real coefficients ([8], [11]).

2.4 *Numerical integration*

In numerical analysis, there is an entire family of methods dealing with numerical integration. Numerical integration finds its applications in domains like embedded systems because in this category of applications the integrand f may be known only at certain points, such as in data obtained by sampling.

But even if an explicit formula for the integrand f is given, it may be impossible to compute the integral exactly because a primitive function cannot be found ([7]).

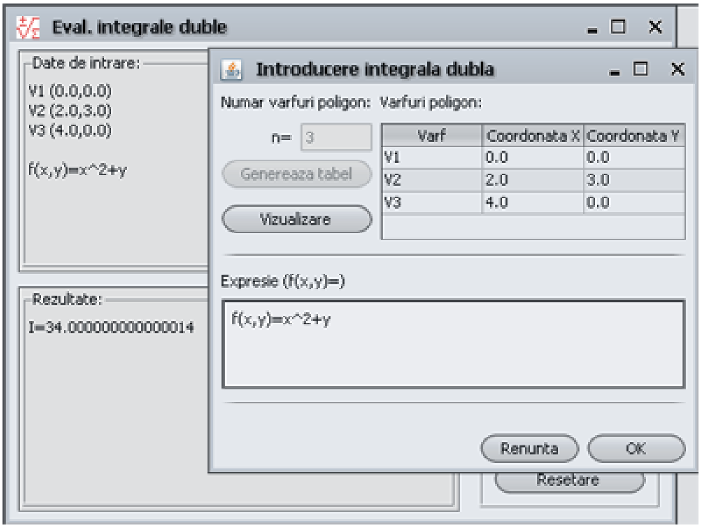

Figure 4. *Representation of the integrability domain for a double integral (Romanian version)*

The methods included in this chapter are:
- Newton quadrature method for evaluating simple integrals (with a given accuracy);
- a method for evaluating double integrals over measurable convex domains with polygonal boundary.

2.5 *Cauchy problems*

Differential equations can be interpreted as a mathematical equation characteristic to an unknown single or multi-variable function. Differential equations represent a headstone in engineering, mathematics, physics and other disciplines. There are several ways of studying differential equations from a mathematical point of view, mainly revolving around their solutions. However, only the simplest differential equations admit solutions given by explicit formulas. Most of the properties of the solution of a differential equation can be determined without finding their true form. The study of stability of differential equations is known as stability theory ([12], [14]).

The methods included in this chapter are:
- Euler method for the solving of a Cauchy first order problem;
- a Runge-Kutta fourth order method for solving Cauchy first order problems;
- a Runge-Kutta fourth order method for solving high order differential equations.

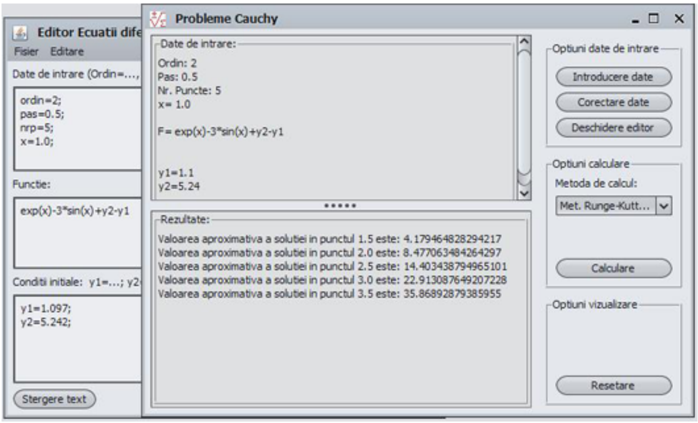

Figure 5. *Numerical solving of I-st order initial value problems, by a Runge-Kutta method, using the Romanian version*

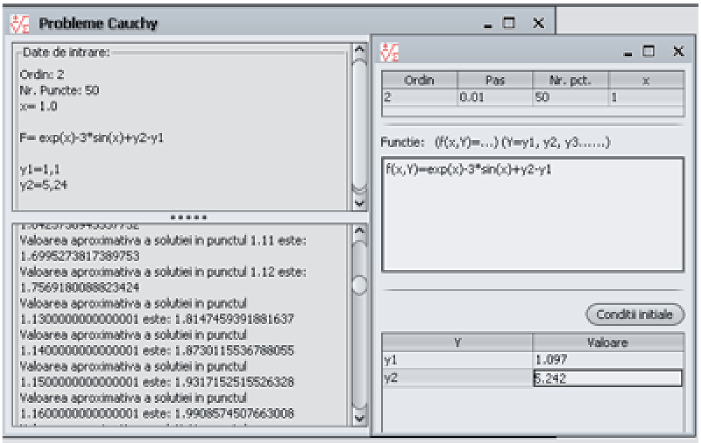

Figure 6. *Superior order differential equations processing (Romanian version)*

3. **Optimization in the new Numerical Engineering Software**

Optimization in Numerical Engineering Software represented a principal feature. Optimization has revolved around two main directions: refining basic methods and developing new ones according to technical needs.

By refining basic methods we understand to take into account all the particular cases in order to make the methods able to process larger categories of data with better accuracy. One such example is the Danilevski method for the calculations of eigenvalues and corresponding eigenvectors which runs on all particular cases.

The refining of basic methods can also mean the ability to impose the precision of calculation, which we implemented in most of our methods. In the new version of NES a number can be represented with a precision of up to

15 decimal places. Nowhere is that fact more obvious than in the calculation of an eigenvector corresponding to a certain real eigenvalue. Each eigenvalue is represented with a precision of up to 15 decimal places.

Another significant improvement we made in the optimization of our program was implementing the ability to impose the degree of the interpolation polynomial in the Lagrange and Newton's interpolating method ([13]).

We also covered all particular cases for the LR method for the computation of the eigenvalues of a real square matrix.

In order to expand the program's applicability we were faced with the hard task of exploring difficult subjects in numerical analysis ourselves. The most demanding section in our program was, without a doubt, the Cauchy Problems chapter. The implemented numerical algorithm for the numerical resolution of a Cauchy first order problem allows us to impose the precision of the results. In consequence, even though the numerical method used is a Runge-Kutta fourth order one, the accuracy of the approximate values of the exact solution depends on the user. Concerning the case of high order differential equations, we obtained an efficient numerical algorithm, based on a Runge-Kutta fourth order method and having a minimal number of input data and also a minimal computational cost.

A new presence in Numerical Engineering Software is the implementation of the new Militaru method for computing the extreme eigenvalues for a real symmetric matrix, with a given precision ([9]). The algorithm avoids the determination of the coefficients of the characteristic polynomial of the given matrix, or the use of similarity transformations, with the purpose of eliminating the intermediate stages of calculation which lead to numerical instabilities, contributing in the decrease of the amount of work.

4. **Updates in the new version of Numerical Engineering Software**

As we stated before in the present report, as of March 2009, the Numerical Engineering Software project has considerably grown in size to reach a new level of development. Improvements were made starting with the interface and ending with the overall number of methods, including several of the optimizations presented in the previous section.

The first improvement we made and is now a part of the new version was the English version, followed very soon after by the French version. This also justifies the appearance of the language drop-down menu on the introductive page of the project. The default value is "English", however, we strongly recommend that the Romanian version be used in order to appeal to as many users as possible.

The second improvement we made in matters of interface was the introduction of separate windows of calculation for the solving of linear systems having a tridiagonal or a pentadiagonal matrix of coefficients.

Other improvements are represented by the ability to impose the degree of the approximation polynomial in the Lagrange and Newton interpolation method. Changes in this chapter also include better error control and the automatic sorting of data for an optimal process data.

In the Numerical Integration chapter, in the section concerning the numerical approximation of double integrals, we introduced a sequence for sorting the vertices of the polygon which represents the boundary of the integrability domain so that the user doesn't need to take into account the succession of the vertices within the geometrical figure when introducing their coordinates (which is the case for polygons having a high number of vertices). In consequence, the polygon vertices can be given randomly, only by their coordinates, the software sorting these points in such a way that they form the geometrical figure corresponding to the integrability domain.

New methods introduced in the new version of Numerical Engineering Software are:
- Militaru method for the calculation of extreme eigenvalues of a symmetric matrix;
- Cubic Spline 1 (cubic spline approximation with free boundary);
- Cubic Spline 2 (cubic spline approximation with imposed boundary);
- a numerical method for solving high order differential equations (based on a Runge-Kutta fourth order method);
- LR method for the calculation of eigenvalues and the eigenvectors corresponding to real eigenvalues, covering all particular cases.

References

1. Burden, R.L., Faires, J.D., *Numerical Analysis*, Brooks Cole Ed., 2004.
2. Ciarlet, P.G., *Introduction à l'Analyse Numérique et l'Optimisation*, Editions Masson, Paris, 1990.
3. Chatelin, F., *Spectral Approximation of Linear Operators*, Academic Press, New York, 1983.
4. Demidovici, B., Maron, I., *Éléments de Calcul Numérique*, Editions Mir Moscou, 1973.
5. Ebâncă, D., *Metode numerice în algebră*, Editura Sitech, Craiova, 2005.
6. Higham, D.J., Higham, N.J., *MATLAB Guide*, The second edition, SIAM, 2005.
7. Leader, J.J., *Numerical Analysis and Scientific Computation*, Addison-Wesley, 2004.
8. Militaru, R., *Méthodes numériques. Théorie et applications*, Editura Sitech, Craiova, 2008.
9. Militaru, R., *On the Newton's Iterative Method for the Characteristic Equation of a Real Symmetric Matrix*, IEEE Computer Soc., Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2006), 175-180, 2006.
10. Militaru, R., Călin, L.A., Călugăru, G., Georgescu, A., *Numerical Engineering Software*, "Analele Universităţii Spiru Haret, Seria Matematică-Informatică", Bucureşti, 4, 39-46, 2008.
11. Philips, G., Taylor, T., *Theory and Applications of Numerical Analysis*, Academic Press, 1999.
12. Popa, M., Militaru, R., *Analiză Numerică*, Editura Sitech, Craiova, 2003.

13. Popa, M., Militaru, R., *Metode numerice – algoritmi şi aplicaţii*, Editura Sitech, Craiova, 2007.
14. Press, H.W., Teukolsky S.A., Veterling T.W., Flannery B.P., *Numerical Recipes*, The third edition, Cambridge University Press, 2007.
15. Ueberhuber, C.W., *Numerical Computations. Methods, Software and Analysis*, Springer-Verlag, 1997.