THE ANNALS OF "DUNAREA DE JOS" UNIVERSITY OF GALATI
FASCICLE III, 2010, Vol.33, No.2, ISSN 1221-454X
ELECTROTECHNICS, ELECTRONICS, AUTOMATIC CONTROL, INFORMATICS

A NEW NUMERICAL CALCULUS SOLVER – NUMERICAL ENGINEERING SOFTWARE

Romulus MILITARU*, Liviu Adrian CĂLIN,**
George-Cristian CĂLUGĂRU*, Adrian-Lorel GEORGESCU*****

**Department of Applied Mathematics, University of Craiova*
(e-mail:militaruromulus@yahoo.com)
*** Faculty of Mathematics and Computer Science, University of Craiova*
(e-mail: adi.calin.nds@gmail.com)
**** Faculty of Automation, Computers and Electronics, University of Craiova*
(e-mail:calugaru.george.nds@gmail.com, georgescu.adrian.nds@gmail.com)

Abstract: The project presented in this paper, entitled Numerical Engineering Software is an integrated solution which contains a various range of numerical methods for the solving of diverse calculations and mathematical approximations from science and engineering. The environment consists at the current stage of development of five chapters: Matrix Algebra, Polynomial Approximations, Roots of Equations, Numerical Integration and Cauchy Problems. The program works through dedicated application windows characteristic to each operation that can be performed and it has the advantage of an accuracy of the results, imposing by the use.

Keywords: linear systems, eigenvalues and eigenvectors, approximations, numerical cuadrature and cubature, Cauchy problems, mathematical software.

1. INTRODUCTION

The project presented in this paper is called Numerical Engineering Software (NES) and is a numerical calculations utility software developed in Craiova by a miscellaneous team from the Faculty of Automation, Computers and Electronics, the Faculty of Mathematics and Computer Science and the Department of Applied Mathematics, all belonging to the University of Craiova.

The project development started from the idea of creating a computer program that can be used in laboratories and also as a guide for students and MBA's alike through its user-friendly interface, helping them to achieve viable results in the smallest amount of time possible and with the minimum effort possible.

The program is achieving new stages of development very fast and is therefore considered a very dynamic solution.

The team of developers has sought to set the basic standards in portability and flexibility for the program to be of real support to the segment intended for.

We consider these basic standards to be in the lines of: possibility of imposing precision of calculus, the special dedicated application windows for every problem that can be solved with the project and the text editor implemented in every section of the program which gives the user the possibility to saves the processed data in special files with the ".nes" extension.

Numerical Engineering Software is also a cross-platform application being able to perform in

This paper was recommended for publication by Adrian Filipescu

operating systems such as Windows, Linux and Solaris.

The current version of the program consists of five chapters: Matrix Algebra, Polynomial Approximations, Roots of Equations, Numerical Integration and Cauchy Problems.

This level of development has been achieved by successive research and implementation stages starting from a nucleus called Numsoft, a rough version, very inflexible, with no optimization stages implemented, to the current day Numerical Engineering Software, (Militaru, *et al*., 2009).

The program was thought of as Romanian interface software, but in order to appeal to more users and make technical terminology familiar to as many users as possible, versions in English and French were implemented successfully.

The applications which can find their solutions through the Numerical Engineering Software are as follows:

- calculation of the inverse of a real square matrix, of the solution of a linear algebraic system, of the characteristic polynomial of a real square matrix, of the eigenvalues and the corresponding eigenvectors; additional matrix operations include determinants calculations and matrix transformations;

- approximation of the value of a function depending on a given set of data points for which its values is known and approximate graphic profile visualization;

-evaluation of the roots of a polynomial;

-estimation of the value of simple and double integrals;

-numerical solving of first order initial value problems for ordinary differential equations and high order differential equations, or for systems of ordinary differential equations;

The complete list of methods through which these problems are solved will displayed and presented further in this report.

The input data can be inducted in Numerical Engineering Software with the benefit of usual basic mathematic functions: sine (sin(variable)), cosine (cos (variable)), tangent (tan(variable)), cotangent (ctg (variable)), exponential function (exp(variable)).

Although the research for Numerical Engineering Software was extensive and continues to bring about more challenges, the need for optimization was abruptly felt in a previous stage of development. The session of optimization of the project was implemented successfully and involves several innovations which improve the complexity and viability of Numerical Engineering Software.

A section of this report will de dedicated especially for the detailing of the results we obtained in matters of optimization.

The development of Numerical Engineering Software is far from over and the team of developers has a distinct set of guidelines which will continue to follow in order to supply a larger number of users with solutions to a greater range of problems.

The current directions of research followed in the development of Numerical Engineering Software include: finalization of the differential equations chapter and the optimization of the methods included in this chapter, finalization of the linear algebra chapters, the study of numerical methods for non-linear types of problems (e.g. nonlinear systems), improvements to the graphic approximations section in visualization and the implementation of 3D graphs visualization and improvements of the text editor.

2. LIST OF NUMERICAL METHODS INCLUDED

Numerical Engineering Software contains a various range of numerical methods for the numerical solving of a large area of mathematical and technical problems. Selection criteria for the numerical methods implemented where established early on in order to avoid redundancies in the project and keep the computational cost as low as possible.

The most important criterion is that of computational cost. The optimization procedures can be seen as effective if the initial computational cost decreases visibly.

Another criterion important in the selection of numerical methods is that of compatibility among methods.

The following is a list of the numerical methods already implemented in Numerical Engineering Software by chapters.

1.1. Matrix algebra

Matrix algebra is very important in all technical calculations, especially those belonging to the engineering field. Thus, in systems engineering the basic science that studies the properties of the systems is called systems theory. A system is defined as a series of abstract entities combined through interdependence relations. There are many types of systems: continual (where the system measures are defined on intervals), discrete (where at least one measure is given as a collection of data points), with distributed parameters (defined by partially derived

equations) etc. The most important property of a linear system is that of stability. For non-linear systems, stability is not a system property. Stability criteria can be algebraic or frequency related. The algebraic criteria are based on processing the coefficients of the characteristic polynomial. For a continuous system, algebraic criteria are represented by the Hurwitz criterion and the Routh table, which are based on solving determinants. Discrete time systems can be studied with the Schur-Kohn criterion or the Jury criterion which are also based on the solving of determinants. Other system properties based on the solving of determinants are the controllability and observability properties.

Controllability is the system property represented by the probability of a system to evolve from a state to another with a given input and output.

Observability is a system property represented by the probability of reconstituting a state of the system based on the knowledge of the input and output of the system. (Ionescu, 1987)

Linear systems are very important in mathematics and engineering. In electrical engineering linear systems can be used to represent electrical circuitry in complex electrical circuits.

Equilibria can be studied with the help of the signs of the eigenvalues belonging to the linearization of the equations about the equilibria. That is to say, for every equilibrium point the Jacobian matrix must be evaluated, and then, after finding the resulting eigenvalues, the equilibria can be categorized. Then the behavior of the system in the neighborhood of each equilibrium point can be determined from the quality point of view, (or even quantitatively determined, in some instances, by finding the eigenvector(s) associated with each eigenvalue).

An equilibrium point is hyperbolic if none of the eigenvalues has the real part zero. If all eigenvalues have negative real part, the equilibrium is a stable node. If at least one has a positive real part, the equilibrium is an unstable node. If at least one eigenvalue has negative real part and at least one has positive real part, the equilibrium is a saddle point. If all eigenvalues are identical null we have an equilibria line.

In computer science, one of the most important applications of matrix calculations is the Page Rank algorithm used by Google to supply the order of the results of a search. The algorithm is based on the processing of large stochastic matrices.

The methods included in the first chapter of NES are (Burden, *et al*., 2004, Chatelin, 1983, Demidovici, *et al.*, 1973, Mellor, *et al*., 2004, Militaru, 2008, Popa, *et al*., 2010):

-Gauss elimination method of inverting a matrix;

-iterative method of inverting a matrix;

-iterative Seidel-Gauss method for systems of linear equations;

-iterative Seidel-Gauss method for sparse matrix linear systems;

-LR factorization for the solving of a linear system;

-LR factorization for tridiagonal matrix systems;

-LR factorization for pentadiagonal matrix systems;

-Fadeev method for determining the characteristic polynomial of a real square matrix;

-Danilevski method for obtaining eigenvalues and eigenvectors of a real square matrix;

-LR method for the calculation of eigenvalues of a real square matrix (including the particular cases);

-Militaru method for the estimation of the extreme eigenvalues of a symmetric real matrix (based on successive approximations), (Militaru, 2006);

-Krylov method for the calculation of the characteristic polynomial of a real square matrix;

-Leverrier method for the calculation of the characteristic polynomial of a real square matrix;

-Jacobi method for the calculation of the eigenvalues of a symmetric matrix;

-QR factorization for solving of a linear algebraic system;

-LR method for matrix transformations;

-QR method for matrix transformations;

-Gauss triangularization procedure for a real square matrix;

-Gauss elimination method with partial pivoting procedure for the numerical solving of a linear algebraic system;

-pivotal condensation method for the calculation of a real square matrix determinant.

1.2. Interpolation and polynomial approximation

One very important aspect of numerical analysis is the study of approximations to functions. Suppose that f is a function defined on some real interval. We now seek some other function p which 'mimics' the

behavior of f on some interval. We say that p is an approximation to f on the given interval.
Generally, such approximations are used because we wish to carry out some numerical calculation or analytical operation involving f, but we find this difficult or impossible because of the nature of f.
For example:
-finding the integral of f over some interval and there may be no explicit formula for such an integral. We replace f by a function p which may easily be integrated;
-evaluating f for a particular point x, especially if f is defined to be the solution of some equation;
- f is given, in the form of a computer procedure, but it is an expensive function to evaluate. In this situation, we are looking for a function p that is simpler to evaluate and produces a reasonable approximation to f.
There are many different ways in which an approximating function p may be chosen, the most commonly used class being the polynomials, according to the Weierstrass theorem. These are easily integrated and differentiated and are well-behaved, in that all derivatives exist and are continuous.

The interpolation procedure has a set of data points as initial conditions and the aim is to determine the value of an intermediate imposed data point.

In technical practice, the data points of an interpolation procedure can be viewed as samples of signal. A sample can be described by a moment in time and the value (voltage) of a signal in that moment in time.

This fact gives us the definition of a signal as a function with a domain and co-domain. The domain is in many situations represented by moments in time.

The process of sampling has its restrictions regarding the sampling frequency. The sampling frequency must be at least twice the size of the signal frequency; in practice, the sampling frequency is tens of times bigger. The bigger the sampling frequency, the better the signal will be reconstructed. (Marin, 2007).

If the sampling frequency is less than twice the signal frequency the aliasing phenomenon occurs which is also known as ambiguity in the frequency domain.

In practice, the continuous signal is sampled using an analog-to-digital converter (ADC).

Measurement instrumentation like oscilloscopes include in their technical specifications the sampling rate measured in samples/second.

For example, for a Tektronix TDS5104B model oscilloscope the sampling rate is 1.25 Gs/s.

The methods included in this chapter (Leader, 2004, Militaru, 2008):

-Lagrange interpolating polynomial;

-Newton interpolating polynomial;

-Free boundary cubic Spline approximation;

-Fixed boundary cubic Spline approximation;

-Discrete least square approximation.

1.3. Roots of equations

A transfer function is represented by dividing the Laplace transform of the output with the Laplace transform of the input in initial null conditions.

A pulse-transfer function is represented by dividing the Z transform of the output with the Z transform of the input in initial null conditions.

This chapter is very important because, to engineers, these roots represent the poles and zeros of a transfer function. The poles represent the roots of the denominator polynomial of the transfer function or of the pulse-transfer function. The poles are used to determine the condition of stability for a continuous or for a discrete system. For a continuous linear time invariant system described by a transfer function the poles have to be inside the left half-plane of the complex plane s (external stability). For a discrete linear time invariant system described by a pulse-transfer function the poles have to be inside the unit disc (external stability). The concept of internal stability is linked with the eigenvalues of a matrix belonging to a state definition, (Marin, 2007).

This chapter also provides good back-up for z-p-k factorizations. (Marin, 2006)

The methods included in this chapter are (Demidovici, *et al*., 1973, Ebâncă, 2005):

-Bairstow method for the calculus of the roots (real or complex) of an algebraic equation;

-Bernoulli root finding method.

1.4. Numerical integration

Numerical integration has been the subject of ongoing research. Nowadays, we can clearly state the links between numerical integration and physical problems.

Numerical integration is used in a wide number of fields for a various range of problems, for example:

-the study of acceptable circuit behavior (electrical engineering);

-find if the concentration of benzene is above or below the toxicity limit at a critical distance from its source (civil engineering);

-how can we infer surfaces from a vector field? (Computer science);

The methods included in this chapter are (Burden, *et al*., 2007, Militaru, 2008):

-Newton quadrature method for evaluating simple integrals;

-numerical cubature method for evaluating double integrals over a measurable convex domain with polygonal boundary.

1.5. Ordinary differential equations

An ordinary differential equation is an equation that involves an unknown function of a single variable, its independent variable, and one or more of its derivatives. Differential equations are the basis for many scientific applications. For example:
-closed loop speed control of a DC motor (industrial engineering);

-controlling sodium chloride waste while making soap (chemistry);

-contracting a cylinder to shrink fit a hub (mechanics);

Related concepts include delay differential equations (DDE), stochastic differential equations (SDE) and differential algebraic equations (DAE).

Differential equations represent a different concept from difference equations even though their theories are closely related.

In systems theory, ordinary differential equations with constant parameters are used to describe continuous linear time invariant systems, the most easily described systems. For systems with a variable structure, the differential equations have the right member discontinuous. The concept of difference equations is used to describe discrete time systems. (Marin, 2007)

In practice, most systems used are non-linear (e.g. logical systems), but their behavior can sometimes be approximated well enough through linear models.

The methods included in this chapter are (Militaru, 2008, Popa *et al*., 2010, Philips, *et al*., 1999, Press, *et al*., 2007):

-Euler method for the numerical solving of a Cauchy I-st order problem;

-a Runge-Kutta 4-th order method for the numerical solving Cauchy I-st order problems;

-a Runge-Kutta 4-th order based method for the numerical solving of high order differential equations;

-a Runge-Kutta 4-th order based method for the solving of systems of differential equations;

3. THE INTERFACE OF NUMERICAL ENGINEERING SOFTWARE

The user interface of Numerical Engineering Software is composed of special dedicated application windows for every problem that can be numerically solved through the program.

These application windows emerged from the need to gain flexibility and the need to keep the restrictions of the methods on hand at all times.

Most of the methods benefit from the possibility of imposing the accuracy of the results.

For the processing of large amounts of data, a text editor has been implemented with the possibility of saving input data to be processed at a later time, files with the special extension ".nes".

The most significant points of the interface of Numerical Engineering Software are presented in the figure complete with sets of input data in order to underline all the capabilities and aspects of the interface.

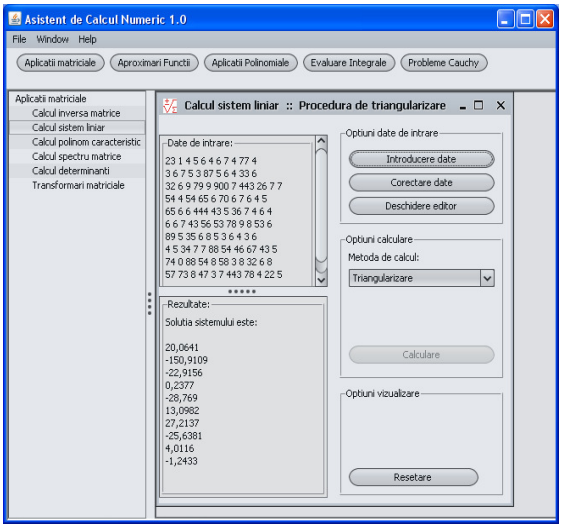


Fig.1. The numerical solving of a linear algebraic system window using the Romanian version of NES.

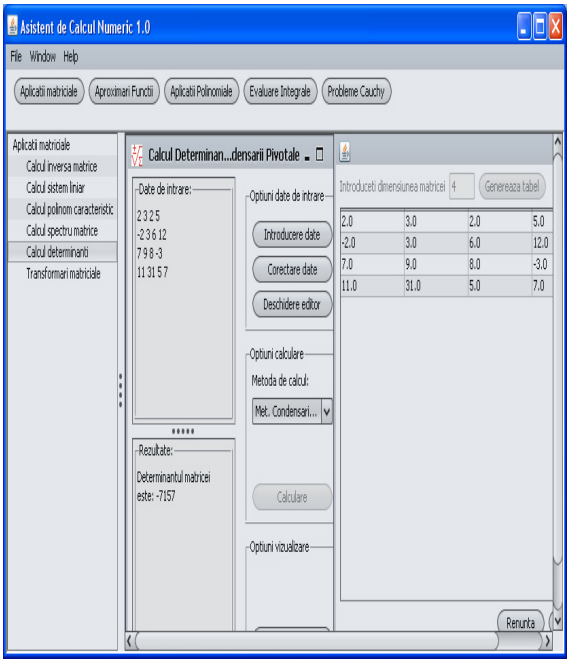


Fig.2. The calculus of a determinant using the Romanian version of NES.

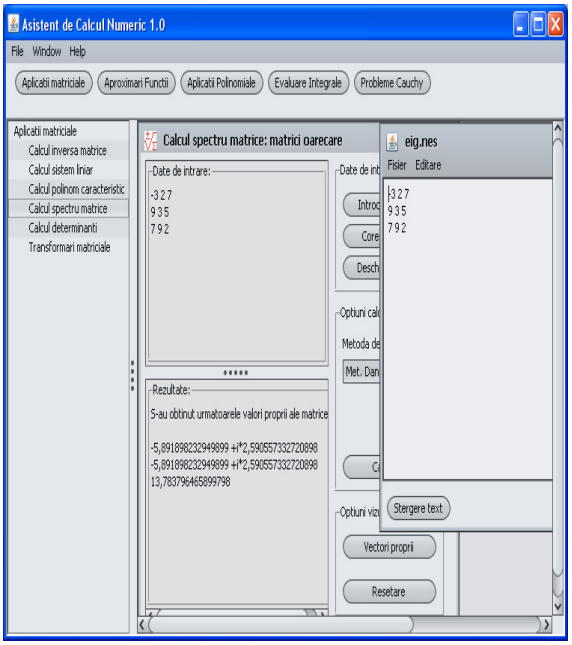


Fig.3. The evaluation of the eigenvalues of a square real matrix, using a ".nes" file for importing data, (Romanian interface).

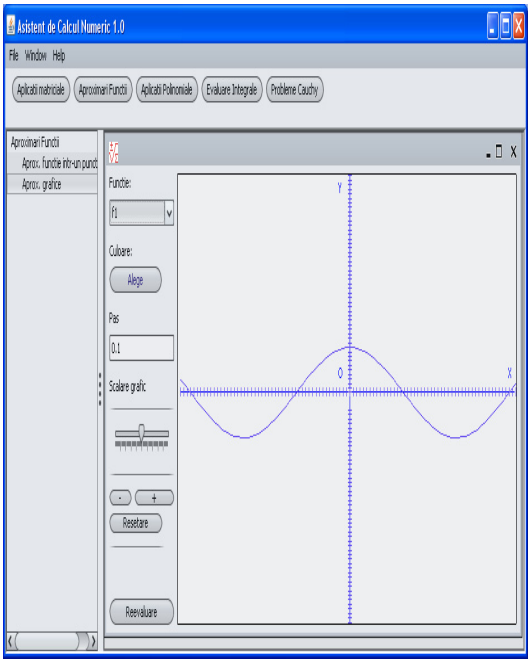


Fig.4. The approximate graphic visualization of a cosine function, using the Romanian interface.

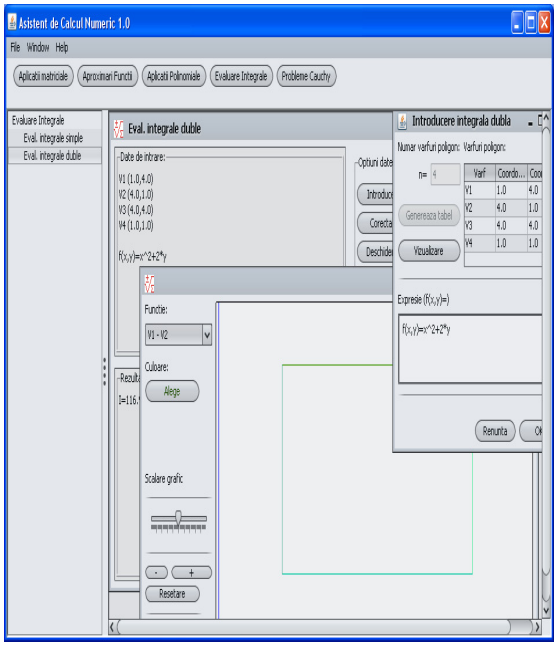


Fig.5. The numerical evaluation of a double integral and the visualization of the domain of integration, using the Romanian version of NES.

Fig.6. A first order initial value problem, numerically solved, using a Runge-Kutta 4-th order method, in the Romanian version of NES.

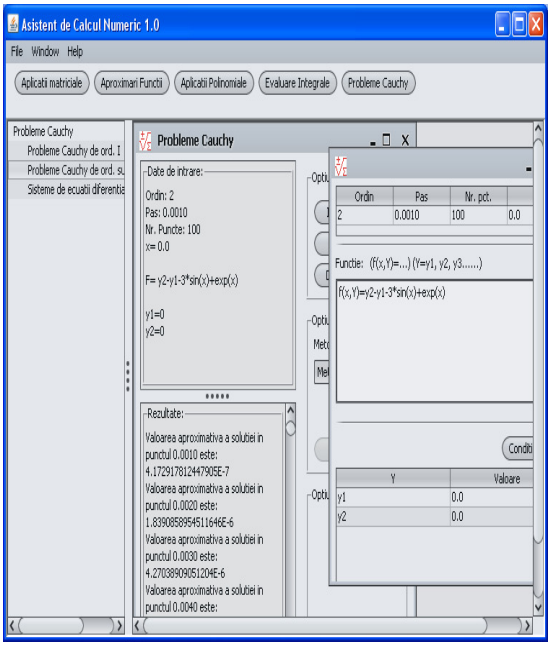
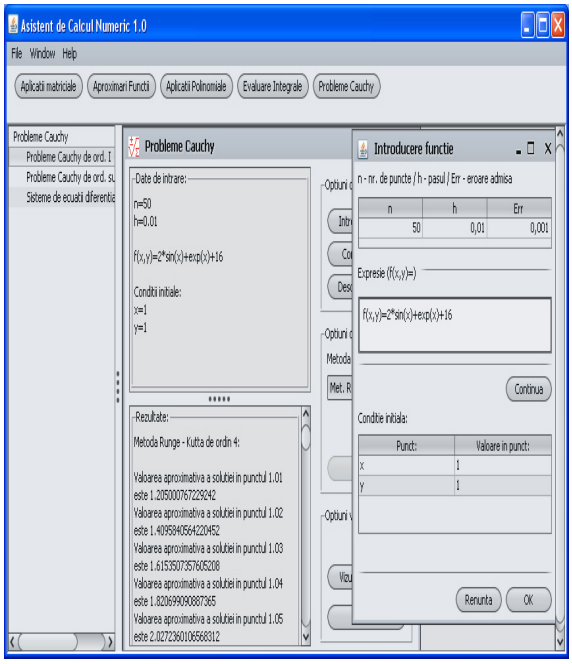


Fig.7. The numerical solving of a Cauchy high order problem using the Romanian version of NES.



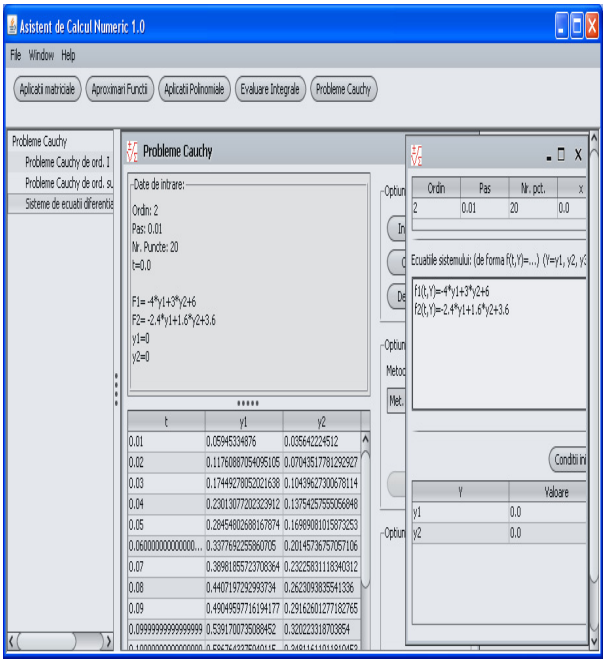


Fig.8. A system of differential equations, numerically solved, using a Runge-Kutta 4-th order method, in the Romanian version of NES

4. OPTIMIZATION SEQUENCES IN NUMERICAL ENGINEERING SOFTWARE

Optimization played an important role in the development of Numerical Engineering Software. The optimizations we implemented help us further the development of the project one inch nearer to completion.

First steps of optimization started with the development of program sequences in order to treat all particular cases of a method (e.g. Danilevski method for the calculus of eigenvalues and corresponding eigenvectors) and ran until the development of new numerical methods (e.g. Militaru method for the evaluation of the extreme eigenvalues of a real symmetric matrix).

The most complex problems of optimization are listed below:

1. Initial value problems for high order differential equations are solved numerically through a compact procedure based on a Runge-Kutta 4-th order method, depending only on the order of the equation, the analytical expression of the highest derivative appearing in the equation and the imposed integration step size introduced by the user. The case of a system of ordinary differential equations is numerically solved by a similar procedure.

2. Runge-Kutta methods (I and IV order) allow the display of the approximate values of the exact solution belonging to an initial value problem for ordinary differential equations, with an integration step size selected by the user. The accuracy of the result can be imposed by the user by inducting the precision of the calculations.

3. The Danilevski method determines the coefficients of the characteristic polynomial, eigenvalues and eigenvectors for any real square matrix. All the particular cases are covered, the algorithm having a minimal computational cost.

4. The LR method allows the determination of the eigenvalues of a real square matrix. The algorithm gives the possibility of working with a given precision. It is complete with all particular cases and has a minimal cost of computation.

5. The Militaru method allows the numerical approximation of the extreme eigenvalues of a real symmetric matrix with a given precision. 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 often lead to severe numerical instabilities. This algorithm also benefits from an optimum cost of computation (Militaru, 2006).

6. Lagrange and Newton interpolating polynomials are used to determine the value of a point within a given set of data points. Optimization of these algorithms in Numerical Engineering Software consists of imposing precision of evaluation, (Militaru, 2003). If the imposed precision can't be reached, the program delivers the best results for the maximum reachable precision. Thus, the algorithm exploits better the results of calculus, contributing to a decrease in the amount of work involved, respectively the computational cost. One can also visualize the approximate profile of the function with an option for scaling the graph and several other appearance options.

7. Numerical cubature method for evaluation of double integrals contains a procedure for sorting the vertices of the polygon which represents the boundary of the domain of integration 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 real domain of integration.

5. NUMERICAL EXAMPLES

1.6. Matrix Algebra

1. For the methods dealing with the solving of a linear system, the following example can be taken into consideration:

$$(1)\ \begin{cases} 5x+2y+z=12 \\ 5x-6y+2z=-1 \\ -4x+2y+z=3 \end{cases}$$

The system matrix is:

$$A=\begin{pmatrix} 5 & 2 & 1 \\ 5 & -6 & 2 \\ -4 & 2 & 1 \end{pmatrix}$$

and the vector of the free terms is given by:

$$b = \begin{pmatrix} 12 \\ -1 \\ 3 \end{pmatrix}$$

Results obtained by applying the triangularization method, LR Doolitle factorization and QR factorization are:

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Results also verified by Matlab.

On this set of input data, the Seidel-Gauss method cannot be applied because the matrix is not diagonally dominant.

2. We are looking to evaluate the eigenvalues of the matrix:

$$(2)\ A = \begin{pmatrix} 4 & -1 & 4 \\ 0 & 5 & -4 \\ 1 & -3 & 5 \end{pmatrix}$$

Using the NES environment and applying the LR method with a given accuracy equal to 10^{-5}, the following results are obtained:

$$\begin{pmatrix} 9 \\ 4 \\ 1 \end{pmatrix}$$

This solution is also confirmed by Matlab.

3. Given a symmetric matrix $A \in \mathrm{R}^{5\times 5}$, obtained from applications based on the finite element:

$$(3)\ A = \begin{pmatrix} 5 & 0 & 0 & 1 & 2 \\ 0 & 4 & 3 & 0 & 6 \\ 0 & 3 & 7 & 8 & 1 \\ 1 & 0 & 8 & 9 & 0 \\ 2 & 6 & 1 & 0 & 10 \end{pmatrix}$$

The purpose is to approximate the extreme eigenvalues of matrix A with an imposed accuracy ε.

The extreme eigenvalues, evaluated with an accuracy of 4 exact digits, supplied by Numerical Engineering Software: 17.1644 and -1.5981.

The extreme eigenvalues of a real symmetric matrix were obtained using the newly developed Militaru method.

This solution is also confirmed by Matlab.

1.7. Polynomial Approximations

1. Considering the following table:

x	0	0.1	0.2	0.3	0.4	0.78	1.33
$f(x)$	-1	-0.620	-0.283	0.0066	0.2484	0.677	-0.2306

We intend to approximate f (0.155) and f (0.947) with a given accuracy ε.

Observation: The exact solution is:

$$\begin{cases} f(0.155) = -0.4299082 \\ f(0.947) = 0.6005443 \end{cases}$$

Using NES environment, with a given accuracy equal to 10^{-4}, the Lagrange interpolation give us the following approximations:

$$\begin{cases} f(0.155) = -0.42985652 \\ f(0.947) = 0.59512606 \end{cases}$$

1.8. Numerical Integration

1. Considering the following defined simple integral:

$$(4)\ I = \int_0^2 x^2 e^{-x^2} dx$$

We are looking to evaluate the exact value with the accuracy 10^{-9}.

Observation: The exact solution with 8 exact digits is $I = 0.42272505$

Using Numerical Engineering Software, with a given accuracy equal to 10^{-9}, the obtained solution is:

$$I = 0.422725056$$

2. Given the following double integral:

$$(5)\ I = \iint_D \frac{x}{1+y^2}\,dxdy$$

D is a domain defined by the following vertices: $V_1(3;1)$, $V_2(4;1)$, $V_3(5;3)$, $V_4(4;4)$, $V_5(3,3)$.

The exact solution is: 2.70391

Using Numerical Engineering Software the approximation of the exact solution within the accuracy $3\cdot 10^{-2}$ is: $I = 2.6879365577656067$

1.9. Cauchy Problems

1. Given the first order initial value problem

$$(6)\begin{cases} y' = y/t - (y/t)^2, & 1 \le t \le 4 \\ y(1) = 1 \end{cases}$$

we are looking to evaluate the approximate values of the exact solution $y = y(t)$, for the points $t_i = 1 + 0{,}1\cdot i,\ \ i = \overline{1,14}$

The exact solution is:

$$(7)\ y(t) = t/(1+\ln t)$$

Using NES environment, the Euler method for an imposed accuracy equal to 10^{-5}, give us the following approximate values:

Point	Approximate value
1.1	1.0042743048159943
1.2	1.0149405491243457
1.3	1.0297992526949782
1.4	1.047517777120264
1.5	1.0672450924521115
1.6	1.088414675681606
1.7	1.1106365322201297
1.8	1.1336346887960853
1.9	1.1572021039512166
2.0	1.1812004212302614
2.1	1.2055142185346244
2.2	1.2300643818508497
2.3	1.2547855604872606
2.4	1.2796269551511412

2. Given the II-nd order Cauchy problem

$$(8)\begin{cases} y'' = 2ty' + 2y - 4t, & t \in [0;2] \\ y(0) = 1 \\ y'(0) = 1 \end{cases}$$

we are looking to evaluate the approximate values of the exact solution $y = y(t)$, for the points $t_i = 0{,}2\cdot i,\ \ i = \overline{1,10}$

The exact solution is:

$$(9)\ y(t) = t + \exp(t^2)$$

Using NES environment, the approximated values computed by a Runge-Kutta fourth-order method are the following:

Point	Approximate value
0.2	1.240565018308538
0.4	1.5731396042263521
0.6	2.0326860155547255
0.8	2.69531988137292
1.0	3.716450656939858
1.2	5.417600108796136
1.4	8.493732889791163
1.6	14.525384513587909
1.8	27.312833107587746
2.0	56.553054865666326

CONCLUSIONS

The Numerical Engineering Software was created to be of real support to any person that needs to find or approximate solution to a technical problem through numerical calculus.

As for its utility, the Numerical Engineering Software has many solutions to applications both for engineers and mathematicians alike, its usage requiring no prior programming knowledge.

Every person working in a technical field can appreciate the benefits of software with an easy to use interface and the fact that almost everywhere the developers have implemented the possibility of imposing the user's accuracy of calculation. This performance of calculation with any given precision is one of the strongest assets of Numerical Engineering Software.

The implementation of numerical methods focusing on the differential equations chapter of mathematics is essential for engineers and the project personnel wishes to exploit this chapter further and give it its final touch as soon as possible.

In conclusion, the development of Numerical Engineering Software follows its regular course, but is far from over.

The team of developers can implement many other numerical methods in Numerical Engineering Software, thus offering the possibility of finding solutions for more problems still, with an interesting diversification in the domains the program wishes to explore (e.g. automation characteristic modules will be implemented in the nearby future).

We hope Numerical Engineering Software will prove itself to be of real value to local academics.

REFERENCES

Burden, R.L., Faires, J. (2004), *Numerical Analysis*, Brooks Cole.

Ciarlet, P.G. (1990), *Introduction à l'Analyse Numérique et l'Optimisation*, Ed. Masson, Paris.

Chatelin, F. (1983), *Spectral approximation of linear operators*, Academic Press, New York.

Demidovici, B., Maron, I. (1973), *Elements de Calcul Numérique*, Ed. Mir Moscou.

Ebâncă, D. (2005), *Metode numerice in algebră*, Editura Sitech, Craiova.

Ionescu, G., Ionescu, V., s.a (1987), *Automatica de la A la Z,* Ed. Stiinţifică şi Enciclopedică, Bucureşti.

Leader, J.J. (2004), *Numerical Analysis and Scientific Computation*, Addison-Wesley.

Marin, C., Petre, E., s.a. (2006), *System theory problems*, Editura Sitech, Craiova.

Marin, C., Popescu, D. (2007), *Teoria sistemelor si reglare automata*, Editura Sitech, Craiova.

Mellor-Crummey, J., Garvin, J. (2004), *Optimizing sparse matrix vector product computations using unroll and jam*, International Journal of High Performance Computing Applications, 18(2), pg. 25-236.

Militaru, R. (2008), *Méthodes Numériques. Théorie et Applications*, Ed. Sitech, Craiova.

Militaru, R. (2006), *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), pg. 175-180.

Militaru, R. (2003), *A polynomial interpolation algorithm for estimating a numerical function*, Annals of University of Craiova, Math. Comp. Sci. Ser., Volume 30(2), pag. 1-7.

Militaru, R., Călin, L.A., Călugăru, G.C., Georgescu, A. (2009), *Efficient Computer Assisted Numerical Calculus through Numerical Engineering Software*, Proceedings of the Conference on Applied and Industrial Mathematics CAIM 2009, Constanta.

Philips,G.,Taylor,T. (1999), *Theory and Applications of Numerical Analysis*, Academic Press.

Popa, M., Militaru R. (2010), *Metode numerice în pseudocod - aplicaţii*, Ed. Sitech, Craiova.

Press, H.W., Teukolsky, S.A., Veterling, T.W., Flannery, B. (2007), *Numerical Recipes*, 3-rd Edition, Cambridge University Press.