International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:2, No:5, 2008

A First Course in Numerical Methods with "Mathematica"

Andrei A. Kolyshkin

Abstract**—**In the present paper some recommendations for the use of software package "Mathematica" in a basic numerical analysis course are presented. The methods which are covered in the course include solution of systems of linear equations, nonlinear equations and systems of nonlinear equations, numerical integration, interpolation and solution of ordinary differential equations. A set of individual assignments developed for the course covering all the topics is discussed in detail.

Keywords**—**Numerical methods, "Mathematica", e-learning.

I. Introduction

A basic course in numerical methods is given at Riga Technical University (RTU) to second-year students of the Faculty of Computer Science and Information Technology. This is two-credit course with only eight lectures per semester and eight tutorial sessions. Two years ago the decision was made by the Department of Engineering Mathematics of RTU to use "Mathematica" during lectures and tutorial sessions (tutorials were, in fact, replaced by lab sessions). The main problem with this course was very limited amount of time allocated for the study (only 32 academic hours including lectures and tutorials). Pass/fail system is used for some courses given at RTU (this means that usually students write a test in the end of the semester and the results of the test are recorded only as "Pass" or "Fail" and no numerical grade is given). The "Pass/Fail" method is also used for the course in numerical methods. The testing in the past was rather formal taking into account large size of classes (usually we have about 260 students allocated to two sections). Several theoretical questions were asked on the test. Students were also asked to solve some very simple problems which do not require deep knowledge of numerical methods (the main problem was the absence of software packages suitable for numerical computations).

The situation has changed two years ago when RTU obtained licenses for "Mathematica". As a result, the Department of Engineering Mathematics of RTU decided to re-structure some courses that are taught to students in computer science, electrical engineering and communications. Calculus, linear algebra and numerical methods are taught now with the help of "Mathematica". In calculus and linear algebra courses lectures are combined with tutorials and lab sessions where "Mathematica" is used to illustrate the basic concepts of the courses. In numerical analysis course only lectures and lab sessions are used.

In the present paper we discuss the basic principles of the teaching module used at RTU to give a basic numerical analysis course with "Mathematica". A set of individual assignments designed to master the knowledge of numerical methods is discussed in detail.

A. A. Kolyshkin is with Riga Technical University, Riga, LV 1048 Latvia (phone: 371-7089528; fax: 371-7089694; e-mail: akoliskins@rbs.lv).

II. Description of the Course

The course in numerical methods is given to second year students of the Faculty of Computer Science and Information Technology at RTU in the fall after students have completed calculus and linear algebra courses during the first year and also after they have completed a course in basic principles of programming. This fact is essentially used for the course in numerical methods. First, students can see how to solve problems with "Mathematica" which they used to solve on paper (such as solution of systems of linear equations or calculation of determinants). Second, with the help of "Mathematica" instructors can effectively use limited time allocated to the course in order to demonstrate some typical problems that can occur when numerical methods are used (such as divergence of iterative methods or calculation of integrals if the integrand is rapidly changing over relatively small interval). In addition, basic programming skills developed during the first year are used to solve different problems where students are asked to write a few lines of code to solve the problem (for example, to solve a system of nonlinear equations by Newton's method).

Several textbooks discuss numerical methods with "Mathematica" [1], [2]. There are also some online resources available [3], [4]. Collection of e-resources for numerical methods using "Mathematica" is given in [3] where the resources available online are divided into the following two groups: (1) in accordance with the method used (for example, Newton's method for the solution of nonlinear equations) and (2) in accordance with the field (chemical engineering, civil engineering, etc.). Different modules for numerical methods using "Mathematica" are available in ref. [4]. At present there are 39 modules covering such areas as solution of nonlinear equations, interpolating polynomials, numerical quadratures and others. Methods discussed in [3], [4] are explained in detail. For example, Newton's method is defined first by means of an iterative formula, then the conditions of convergence are discussed, finally, "Mathematica" programs are also presented. References [1] – [4] are used as optional sources in our course.

The course includes the following topics: (a) systems of linear algebraic equations (direct and iterative methods of solution); (b) nonlinear equations and systems of nonlinear equations; (c) interpolation and approximation; (d) numerical integration and differentiation; (e) solution of ordinary differential equations (initial value problems and boundary value problems).

Two sets of problems are assigned to each student during the course. The first set consists of 10 problems and is based on numerical solution of systems of linear equations (both direct and iterative methods). The second set has 6 problems which cover such topics as nonlinear equations and systems of nonlinear equations, interpolation, numerical integration and numerical solution of ordinary differential equations. The formulation of the problems is the same for all students. For example, students are asked to solve a system of linear algebraic equations of the form $Ax = B$ by means of an iterative method depending on the parameter τ, but the input data (that is, the elements of the matrices A and B) are individual for each student and depend on the numbers of their student ID's. This means that, depending on the values of the coefficients, for the given value of τ the method can converge or diverge. Thus, student has to adjust the parameter τ using theoretical recommendations in order to guarantee the convergence of the iterative method. Some of the problems from the two problem sets are discussed in the next section.

III. Description of Homework Assignments

The first problem set consists of 10 problems. The objective is to test students' knowledge of numerical linear algebra. The first five problems are rather standard and require the solution of a system of linear equations by Gauss' method or LU-decomposition using "Mathematica" commands **RowReduce**, **LinearSolve**.and **LUDecomposition**. Students are asked to calculate the determinant of a given matrix using "Mathematica" command **Det.** The result should be checked by application of the LU-decomposition of the same matrix. In addition, the inverse of the given matrix should be calculated together with all the eigenvalues. Each student is given his/her own matrix whose coefficients are directly related to the last three digits of his/her ID.

As our experience shows, the remaining problems in the first problem set are more challenging for the students. The reason is that one cannot solve these problems by simply using one "Mathematica" command since the solution should be found by iterative methods. This means that students have to write their own (but rather simple) code to implement the iterative method. In addition, some iterative methods can contain parameters which are used to accelerate convergence of a numerical method. However, for some values of the parameters the method can diverge. As a result, students may need to adjust the parameter in order to guarantee convergence or even select the optimal value of the parameter (if such estimates are known from the theory). These points are illustrated by means of examples.

Example 1. Solve the system $Cx = d$ using factorization method. Compare the result with the solution obtained by means of "Mathematica" built-in function **TridiagonalSolve**. The elements of the matrices C and d are

$$C = \left(c_{ij}\right),\ i = 1,2,...,6;\ j = 1,2,...,6\ ,$$

$$d = (a\ b\ c\ a+b\ b+c\ a+c)^T,$$

$$c_{ii} = -a^2 - b^2 - c^2 - 5,\ i = 1,2,...,6;$$

where $c_{i+1,i} = a+1,\ i = 1,2,...,5;$

$$c_{i,i+1} = b+1,\ i = 1,2,...,5;$$

and a, b and c are the last three digits of student's ID (if one or several parameters a, b and c are equal to zero, then these parameters are replaced by 10). The elements of the matrix C are chosen so that for all a, b and c it is diagonally dominant, that is, factorization method is stable. In order to implemet this method students have to write a few lines of a code. The check of the code is done by comparing the results with the built-in function **TridiagonalSolve**. The problem is considered to be solved if and only if the results of both calculations are identical.

Example 2. Solve the linear system $Px = h$ using the Jacobi method. Comment on the rate of convergence of the method. The matrices P and h are given as follows:

$$P = \begin{pmatrix} a+b+c+1 & a & b & c \\ b & a+b+c+2 & a & c \\ c & b & a+b+c+3 & a \\ a & c & b & a+b+c+4 \end{pmatrix}$$

$h = (c\ b\ a\ 2c+b)^T$. The initial approximation to the solution is given by $x^{(0)} = (0\ 0\ 0\ 0)^T$.

As in the previous example, the matrix P is diagonally dominant for all a, b and c so that the Jacobi method is convergent [5]. However, tne rate of convergence can be rather slow. Students are asked to comment on the reasons for slow convergence.

Example 3. Solve the linear system $Fx = g$ by means of the iteration method

$$x^{(n+1)} = x^{(n)} + \tau(b - Ax^{(n)}), n = 0,1,2,... \qquad (1)$$

The matrices F and g have the form

$$F=\begin{pmatrix} a & b & 1 \\ b & \dfrac{b^2}{a}+1 & \dfrac{2b}{a}-1 \\ 1 & \dfrac{2b}{a}-1 & a^2+b^2+1 \end{pmatrix}, \quad g=\begin{pmatrix} a+b+c+1 \\ a+2b+c+2 \\ c \end{pmatrix}$$

The initial guess is given by $x^{(0)}=(1\ 1\ 1)^T$. The value of the parameter τ is fixed at $\tau=0.05$. In case of divergence adjust the value of the parameter τ to ensure convergence. Compare the number of iterations with the case where the parameter τ has the optimal value.

Method (1) can be used if the matrix F is symmetric and positive definite. It can be shown by means of the Hurwitz criterion that the matrix F is positive definite for all a,b and c.

From our point of view method (1) should be definitely included in any basic course on numerical methods. There are several reasons for that. First, the method is rather simple and only a few lines of „Mathematica" code are necessary to implement the algorithm. Second, the optimal value of the parameter τ is known for this case, namely,

$$\tau_{opt}=\frac{\lambda_{\max}+\lambda_{\min}}{2}, \tag{2}$$

which ensures the highest speed of convergence. Here $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and the smallest eigenvalues of the matrix F, respectively. In addition, one can easily illustrate slow rate of convergence of iterative methods using the estimate of the norm of the residual. During the lectures the instructor shows that for an iterative method of the form

$x^{(n+1)}=Dx^{(n)}+f,\ n=0,1,2,...$

the norm of the residual, $\|r\|$, after n steps is bounded by

$\left\|r^{(n)}\right\|\le\|D\|^n\left\|r^{(0)}\right\|$

Thus, the rate of convergence is directly related to the norm of the matrix D. The rate of convergence is very slow when $\|D\|$ is very close to 1.

The following example during the lecture is used to illustrate the idea.

Consider a linear system of the form

$$\begin{cases} 4x_1+2x_2+14x_3=14 \\ 2x_1+17x_2-5x_3=-101 \\ 14x_1-5x_2+8x_3=155 \end{cases}$$

The coefficient matrix of the system is symmetric and positive definite. The instructor illustrates the solution using method (1) implemented in "Mathematica" with $\tau=0.01$. If the initial guess for x is $x^{(0)}=(0\ 0\ 0)^T$ and $\varepsilon=0.001$ then after 763 iterations the solution converges to the following vector $x^{(763)}=(2.99911,-5.99984,1.00016)^T$. Note that the exact solution is $x=(3,-6,1)^T$

Next, the instructor illustrates the conditions of convergence. An attempt to use $\tau=0.05$ for this problem gives the following result after 35 iterations:

$$x^{(35)}=\left(3.80026\cdot10^{17},-1.51919\cdot10^{17},2.23989\cdot10^{18}\right) \tag{3}$$

The method is clearly divergent. Thus, the instructor states the conditions of convergence of the method: the value of τ should satisfy the following condition

$\tau<\dfrac{2}{\lambda_{\max}}$, where $\lambda_{\max}$ is the largest eigenvalue of the coefficient matrix. This fact also gives an opportunity to the instructor to pay students' attention to the importance of matrix eigenvalues in numerical analysis [6]. Next, the optimal value of τ which is defined by (2) is used for the problem. The method converges to the prescribed accuracy in 474 iterations.

Finally, the instructor should illustrate why so many iterations are needed to obtain the result with prescribed accuracy. It can be shown that for this example the matrix D is defined as follows:

$$D=E-\tau A,$$

where A is he coefficient matrix and E is the 3×3 identity matrix. The rate of convergence is given by $\|D\|_2=\sqrt{\lambda_{\max}(D^TD)}=0.989$. Since the norm of the matrix D is very close to 1, the rate of convergence is so slow.

Essentially students are asked to repeat this analysis for their (individual) problems. Our experience shows that not all of the students understood the concept of convergence. In some cases the instructor received "solutions" similar to (3) – students believed that "computer has obtained the answer"!

Example 4. Find the largest eigenvalue of the matrix F from Example 3 using the power method.

This example is directly related to Example 3 since $\lambda_{\max}$ is needed to specify the interval of the values of the parameter τ for which method (1) is convergent. During the lecture the instructor can also discuss relatively high rate of convergence of the power method (usually $\lambda_{\max}$ is obtained with high accuracy only after a few iterations).

The second problem set consists of 6 problems. The topics include solution of nonlinear equations and systems of nonlinear equations, interpolation, approximation, numerical integration and solution of ordinary differential equations.

Example 5. Find all the real roots of the cubic polynomial $x^3-ax^2+bx-c=0$ (recall that a,b and c are the last three digits of student's ID) with correct five decimal places. Compare the results with the roots obtained by means of the built-in function **FindRoot**.

It is well-known that any cubic polynomial with real coefficents has either one or three real roots. The solution includes the following steps: (a) construct the graph of the polynomial and decide how many roots does the polynomial have; (b) find the initial guesses (from the graph) for all roots; (c) calculate each root using Newton's method.

During the lecture the instructor demostrates the case where the polynomial has three real roots. Therefore, one needs to find three initial guesses for the roots. The instructor should also pay students' attention to the fact that that the initial guesses should be in close proximity to the root (otherwise convergence is not guaranteed). The example of a divergent method is also shown. Both examples were illustrated with animation in order to make the analysis more attractive. Obviously, for some students their problem will have only one real root. It was a surprise to know that some students (whose data were such that only one real root exists) still made all the three steps (following the example shown by the instructor), that is, they used three different guesses (for the single real root!) and presented the idential results for all three roots. This simply shows that the concept of convergence was not well undrstood by all students.

Example 6. Solve the system of nonlinear equations using Newton's method with $\varepsilon = 0.001$ in the region $x > 0,\ y > 0$. The system has the form

$$\begin{cases} \sin\left(\dfrac{x}{a}\right) - y = -0.1 \\ \left(\dfrac{x}{a}\right)^2 + \left(\dfrac{y}{b}\right)^2 = 1 \end{cases}$$

It can be seen that the system has one real solution (since the two curves which geometrically interpret the equations in the system intersect in the first quadrant). One can also reduce the system to a single equation by eliminating one of the variables but students are asked to solve this problem using the method developed for systems of nonlinear equations.

Example 7.

Calculate the intergal $\displaystyle\int_0^T \frac{\sin(c\sqrt{x^2+(a+b)^2})\sin cx}{x\sqrt{x^2+(a+b)^2}}dx$,

where $T = 10, 30, 50, 70, 90.$

Compute the integral $\displaystyle\int_0^\infty \frac{\sin(c\sqrt{x^2+(a+b)^2})\sin cx}{x\sqrt{x^2+(a+b)^2}}dx$.

Compare the result with the exact value of the integral:

$$\int_0^\infty \frac{\sin(c\sqrt{x^2+(a+b)^2})\sin cx}{x\sqrt{x^2+(a+b)^2}}dx = \frac{\pi}{2(a+b)}\sin c(a+b).$$

Comment on the difference between the results. In case of slow convergence use the option „oscillatory" to calculate the integral.

This example shows typical problem in numerical integration if the integrand is oscillatory. Students learn how to identify problems related to slow convergence. In addition, different options available in „Mathematica" can be used to calculate the integral with the given precision.

IV. Discussion

Since "Pass/Fail" grading system is used to evaluate students' performance for the numerical analysis course (no numerical grade is given) the passing grade is obtained if and only if all 16 problems from the two homework assignments are solved. (a kind of take-away exam). A special e-mail address was created by the instructor in order to organize communication with students. Students were asked to send their solutions to the specified e-mail address. The instructor evaluated their solutions by sending his comments to students. If some problems were not solved or were solved incorrectly the instructor sent his feedback and indicated what went wrong and what steps should be taken in order to overcome all the difficulties. All lecture materials and problems solved during lab sessions are available online (including "Mathematica" codes in pdf). The codes are provided in pdf for two reasons: (1) the codes need to be slightly modified in order to be applicable to the solution of homework assignments and (2) we believe that testing and debugging of codes is a useful additional training for future IT specialists.

Students expressed their satisfaction with the approach used by the instructor since it allows them to learn how to overcome problems that occur when numerical methods are used. In addition, they use the opportunity to communicate with the instructor by e-mail and ask questions related to the course.

The teaching module developed at RTU to teach a basic course in numerical methods is discussed in the paper. The approach is based on individual assignments (take-away exam) and extensive use of "Mathematica".

References

[1] D. Dubic, *Numerical and analytical methods for scientists and engineers using Mathematica*. New York: Wiley, 2003.
[2] C.-L. Sandblom, *Practical optimization methods with Mathematica applications*. New York: Springer, 2000.
[3] http://numericalmethods.eng.usf.edu/nbm
[4] http://math.fullerton.edu/mathnews/n2003/Web/NumericalModels.html
[5] G.H. Golub, and C.F. Van Loan, *Matrix computations*. New York: The Johns Hopkins University Press, 1996.
[6] R.A. Horn, and C.R. Johnson, *Matrix analysis*. London: The Cambridge University Press, 1985.