Computer Assisted Mechanics and Engineering Sciences, 15: 143–163, 2008.

Symbolic computations in modern education of applied sciences and engineering

Marcin Kamiński
Chair of Mechanics of Materials, Technical University of Łódź
Al. Politechniki 6, 90-924 Łódź, Poland

(Received August 18, 2008)

A presentation of the modern issues related to the symbolic computing is contained here together with the detailed discussion of its application to the education of various scientific and engineering academic disciplines. The future expansion of the symbolic environment is described here on the basis of their historical and modern developments presentation. As it is shown on the example of the MAPLE system, symbolic computational environments play the very important role in supporting the lectures and the classes in the computer labs. Those environments may be also very useful in teaching basic natural sciences in all those cases, when some algebraic or differential equations appear, must be solved and their results should be precisely discussed. The application of the MAPLE and similar computer systems in the engineering education seems to be unquestionable now and some examples are contained here to show how to improve the lectures and make them very interesting and exciting. The key feature offered by the symbolic computing is the opportunity to discover the knowledge that the students may do by themselves, when they are specifically leaded by the instructors.

Keywords: symbolic computations, computer science, computers in education

1. General introduction

Nowadays, a role of the computer-based learning in all scientific and engineering disciplines and subjects still seems to be decisive and is not limited of course to the symbolic computing programs application. Those computer methods are of course very widely understood and begin with the internet browsing and searching for some specific knowledge sources (the advanced search allows to train logic operators). The teachers and lecturers should focus the students not only on finding the answer for some specific problem but, first of all, on the reliability of the particular internet source. Considering a common practice of finding everything, which is needed to prepare the homeworks at the schools and universities on the relevant webpages, the teachers influence is practically strongly limited to a verification of those sources (without the opportunity of the effective corrections of the net source in its electronic version).

The second process strictly connected to the above is an explosion of the computer software (well packed into the graphical environments). Starting from 3D games through business packages till the scientific advanced programs we can all use, modify or even build any necessary processes flows to complete most of our work to be done on (and mostly) by the computers. Therefore, a role of the modern teaching in this context may be (and really is) limited to a creation of the written or spoken manuals to the specific computer software. In this context the Latin idea that *'repetitorum mater studiorum est'* transforms to the repeated usage of the same program or just the algorithm. Anyway, it may result with no doubt in a complete disability of our children (and us too) to remember correct grammar structures, for instance. So that, the complete computer preparation of the homeworks for the English or history lessons should be strictly prohibited. No one at this moment may prognose the social progress under such circumstances (the written language may be totally left for the computers as well in the nearest future).

The educational aspects of the *'computerization everywhere'* have nevertheless many positive aspects, especially for the basic natural and engineering sciences. The basic and the very clear difference to the previous blackboard methods is the opportunity to (1) present the solution of the particular problem in its final form with its overall verification, (2) compare on the same screen the solutions of the similar or even different problems, (3) study parameter sensitivity of those solutions and, especially (4) make a dynamic visualization of our problem solution. So that we, as the teachers, may pack the complicated (boring for the students) problems together with the necessary knowledge as the attractive colorful boxes, which the children and the students want to open with no delay. It was quite impossible before and must be exploited now as widely as possible. Thanks to such an educational method a large portion of the differential equations appears as a quite natural prolongation of the well-known elementary algebraic equations, where some visible and touchable (on the screen) functions give the final answer.

Most of the lecturers of the computer science courses have wrong feeling that their courses, thanks to the presence of the hardware, colorful software and powerful 3D graphics and animations are so interesting that they need no special attention or separate educational tricks to focus the students mind on this subject. It is true for the several number of teaching hours when the students do not have the ability to use the computer as they want. When they become to be fluent, they start to be bored by the traditional computer science courses, they start to browse the internet in a quite opposite direction, sometimes even to play the games etc. The arguments that the design process or the computer-based analysis are very important, must be introduced in a proper way, sometimes cannot be followed step by step and needs even more attention than the traditional calculations which are frequently ignored by the students. An attractive packaging of any problem is relatively easier here than for the traditional courses, far from the computer methods, but anyway need the academic teachers attention.

There is no doubt that the symbolic software created the brand new opportunities for the lecturers, which are demonstrated and discussed below. Such packages like MAPLE [3], for instance, are present in the regular courses of computer science at many universities, however thanks to the recent progress in symbolic software development the role of this software essentially changed last years. It is reflected by inclusion of the symbolic software into the mathematical handbooks and textbooks [1]. One of the most recent and most apparent educational applications is providing the lectures or even the entire courses in symbolic algebra programs, which can include numerical verification of more important equations together with their straightforward derivation. Such a course may be saved in this package format, so that having this system available at all computer labs, the course can be distributed by email (a lecture by lecture) to all participants to be used during the classes. Let us underline that most of the illustrations are prepared in the symbolic environment of the program MAPLE, however, symbolic systems are nowadays so similar that those illustrations can be translated without any loss to any computer system of this type.

Considering above, we would like to address the following issues in this paper — do the symbolic computer systems offer something qualitatively new, how to teach efficiently the usage of symbolic tools, how to correlate effectively the symbolic computing with the educational needs, how we can modify symbolic programs to carry out the research and the lectures as well as how to interact with symbolic environment software companies. The discussion starts in the next section with the presentation of general aspects of the computer science teaching explaining the basic rules and observations dealing with the computer science academic courses as well as a discussion on the role of internet and games in education of the young people nowadays. The third section contains some introductory comments and the several historical facts about the symbolic computing together with short comments on the future development of the computer algebra systems (CAS). Both next two sections contain the detailed description of the symbolic programs examples that can be used for lower level courses (fifth section) and for the PhD students (sixth section); the educational comments are added on the occasion of each of those examples. The paper is finished with the concluding remarks following the considerations drawn from this work.

2. General aspects of computer science teaching

2.1. Various methodologies of computer science teaching

The computer science and all computer-aided courses (from typical for all the engineering courses to fundamental sciences like applied mathematics) should have the specific educational methods reflecting the fact that they are provided in a computer lab. So that they are based on projects or exercises solved by the students during the classes and completed by themselves at home. The educational strategies are quite different, depending on the university rank, the expectations of the lecturer, the abilities of the students, the course type and the presumed knowledge level of those students. Usually, the following schemes are applied at the computer labs:

1. the students is taught by an observation of the lecturer presentation given during the classes; analogous problems are left for the homework;
2. the student follows on his/her computer the teacher presentation given during the classes step by step;
3. the lecturer explains the entire problem (or some class of the problems further distributed amongst all students) and give a solution algorithm during the presentation or, after a short break, on the blackboard (blackboard explanation seems to be similar to the same algorithm creation by the student on the paper sheet);
4. the lecturer explains the whole problem but the algorithm is left for the students' ideas and initiative - further programs and computer activities during and outside a lab follows this critical point;
5. the lecturer comments the key points of the problem to be solved by students only and gives on the blackboard all necessary equations to solve it;
6. the lecturer explains the key points of this problem and leaves finding of all equations to the students (using their knowledge from previous courses, course books and/or internet sources);
7. the lecturer explains only the whole problem, where the solution, its algorithm, equations to be implemented to find this solution (and possibly its physical interpretation) are the students' job;
8. the lecturer may be the exercises distributor only and then, the students must provide the entire solution (but the problem is generally known from the previous or parallel courses);
9. the student works alone in the computer lab having the problem (or a list of problems to be solved) available in the net or given at the beginning on the paper sheet;
10. the students may work at home, giving the presentation temporarily or at the end of the entire course during some kind of the final graduation.

All those methods can be used in different combinations, where the final decision (to assure the best effectiveness of the course) depends on the lecturer, the students level, the difficulty level of the problems to be solved and on the interaction between the students and the teacher.

Firstly, let us note that the first is a rather wrong way because the lectures become very fast a more or rather less interesting film and the students sitting behind the computers start to do something else. So that most of the knowledge summarized in the presentation must be repeated, which leads undoubtedly to the essential time waste. The second option is not so dangerous, but anyway the mental scheme repeated class by class is that the student can still count on the lecturer, and there is no essential need to do everything by himself; effectively some portion of the exercises is done by the lecturer, which does not belong to this lecturer expectations rather. The third possibility should appear during the classes — at the first time — to show the students the necessity of the good algorithm creation and its optimization — but only once — to avoid the situation that the lecturer will deliver the algorithms for all problems and to introduce the students as soon as possible into the algorithm creation by their own. The situation that the teacher explains in detail the whole problem to be solved together with all necessary equations may appear but only once or twice — the problems should deal with the knowledge given to the students before, during the preceding academic or even high schools courses, so that there is no real need to bring them all equations each time and to create a rule that some more important scientific problems will be explained during the computer science classes and, furthermore, that there is no connection between different science branches. This is not the very recommended way in the age on interconnections between even extremely far scientific disciplines, where computed science could be some platform to make them closer. The next option, where the lecturer comments the most important issues seems to be better approach, but it is good to perform it several times only and not always, of course. The help to the students during the classes seems to be good in most cases, especially to understand the critical points do not belonging to the computer science itself, but, on the other hand, the teacher must assure the students that the computer-aided solution of any problem must include deep understanding of this problem out of the computer and is not a computer game; therefore, this option seems to be case-sensitive. Hence, the next option, when the lecturer explains only the problem and the rest of the solution is left to the students' work seems to be the most justified and the most frequently applied educational scheme. The next options (8, 9 and 10) are provided for better students only, who are believed to work really alone; especially the last option consisting in an almost permanent typical homework should be given for the best student only as an honor.

Whatever and however the lecturer will finally adopt as his approach to the students, it is very useful to remember that the students (a) should be very well motivated (not only by the good grades), (b) must be appreciated (not on every occasion but anyway), (c) may be focused on each problem (even the most boring) by its interesting unusual presentation, (d) they simply like to compete. One can accomplish those goals by treating traditional exercises like the complicated unsolved research problems, so that the students believing that they make the brand new discoveries are strongly associated to the given exercise. Of course, there is no optimal educational methodology independent from many reasons, both predictable and unpredictable, however, treating the students groups statistically, the most recommended way seems to be a continuous transition from the 3rd point to the 8th or 9th point from a lecture to lecture. Then, we start the course from the extensive explanation of almost any activity and, by a systematic elimination of some activities performed or repeated fluently by the students, we tend to the situation when everybody works (including a lecturer) alone at the lab. It is a good moment to underline a very interesting role of the incompletely defined problems we can use for the second part of each course, when the students have been taught basic methods and tricks. The solution process of such problems may be discreetly observed by the teacher since the students to complete this process need to ask for some parameter, assumptions or facts, so that it is possible to make the calendar who asks first and when, to distinguish between better and worse students etc.

Finally, let us remind that there are several ways that cannot be followed according to the opinion of many experienced teachers during the computer science labs and lectures:

1. The problems should not be given as the pictures – especially for the engineering courses – we do not solve pictures but the real problem, where the right scheme creation may be even the most important part of the solution process. Instantaneous graphical presentation of the problems leads to the complete lack of understanding what is really solved.

2. When all the students will get completely different problems they will not interact with each other; it prevents interchanging of the camera-ready solutions between the students (to eliminate this phenomenon we can simply dismiss the mailing tools usage during the classes) but, at the same time, it cancels any interaction of the students, which is important — explanation of the

moment which some students cannot solve or just some questions to the colleague at the left or right needs some understanding of what we really do now.

Let us also remember that:

1. The final result should preferably contain the figures with titles and comments, at least the few sentences commenting the results obtained and their agreement with the initial expectations, the existing knowledge and the common sense; it would be very worthy to demand some parametric studies from the students – it gives a lot of the so-called 'engineering intuition', which extends their imagination of the given project.
2. The entire methodology and our expectations for the problem solution will introduce the students into the scheme, which next may be exploited during the real life problem solutions and can avoid any failures or mistakes during their engineering or research practice.

At the end of this section we can recommend the following problem, which can be solved by the students of many engineering courses using symbolic computer programs and during the computer science classes (if, of course, the theoretical mechanics course was taken before).

Problem: Solve the following equation, give its physical interpretation, make its additional visualization, perform the parametric studies with respect to m, c and k, provide the available animation. What is the physical meaning of the first and the second time derivatives of the solution?

$$m\,\frac{\mathrm{d}^2 x(t)}{\mathrm{d}t^2} + c\,\frac{\mathrm{d}x(t)}{\mathrm{d}t} + kx(t) = f(t).$$

Each student may get here different function $f(t)$ but everyone should: (1) identify this equation as the 2nd order ordinary differential equation, (2) identify its physical meaning as the vibration of the single-degree-of-freedom system, (3) define properly the right hand side function and to introduce the initial conditions (first check point), (4) make a visualization and discussion of the results, (5) give a physical meaning of the velocity and acceleration (the 2nd check point) and finally, (6) produce the animation.

2.2. The role of internet and games in education

Although initially those two subjects were not very closely related to each other, today most of the games are found in the internet thematic websites and, first of all, may have some bad influence during the academic education. The commercialization and the public access to both phenomena of the virtual nature may have some unwanted, unpredictable and even bad influence on the students, so that they need some academic attention.

As it was mentioned in the previous considerations, the internet knowledge sources play very important role in the up-to-date computer-based learning, sometimes even decisive but not always in a positive meaning. According to our observations in the computer labs and in the academic libraries, the electronic sources almost replace written sources, which were traditionally reviewed or at least verified at the printout stage. Now, when the students must solve some problem very fast having completely no or very limited knowledge on some subject, they employ some popular web browser by putting some phrases or their various combinations; it can be for instance www.wikipedia.com. Neglecting the web page address and their inventors and internet providers maintaining this knowledge source, its verification process consists in further solution of the initial problem using this knowledge and presentation of the results to the lecturer. When the lecturer approves this solution once, sometimes twice this web page (or the address) becomes more and more reliable and they are used without any further authorized approvals. This phenomenon is so popular now that some parts of the students populations believe without any doubts in all internet sources even more than in the academic lectures or textbooks. It is not sensible to point out how dangerous it is that the improper facts and improper sources migrate between the whole groups of the students, which can be detected during the final exams, when the students may frequently recall that some specific part of the material has been found in the internet. Sometimes the students by themselves solve some problems supposed to be the subject of the forthcoming exams and put them into the new private (secret) websites. It needs of course further intensive attention of the lecturers and the academic authorities how to prevent this situation. Undoubtedly one may create and recommend simple thematic educational websites, where the entire department may collect all the lectures, presentations together with some solved examples to shorten internet browsing amongst the students and to eliminate the possible errors. It does not mean of course that the internet plays bad role in the educational process, when the lecturers cannot create the thematic internet sources, they are always able to recognize the most wanted addresses by the students and to spend some time during the classes to verify the facts contained on those pages. This problem seems to be rather trivial but the academic practice is that it is usually omitted during the education of computer science almost at all.

This is not a secret that one of the most powerful engines of the computer business development is the computer game industry and that the young people are very sensitive today to those problems playing frequently different games — at home, at the office, at the university and sometimes even during the classes in the computer labs. The power of this phenomenon is quite clear on the mobile phones market and small electronic devices designs like the very modern last days PSPs. It is needless to say that the games (outside mathematical games theory) are not included in any computer science course and, at the same time, that the students try to play during the computer science labs because of an easy access to the reliable and sufficiently fast computer equipment. Neglecting the fact that such an activity means the serious time waste and is inconvenient during academic activity time, there is some much more serious reason to control the time spent behind the games and, first of all, their character. Postponing a typical classification of different games available in the net, the teachers should distinguish between the games having neutral mental influence on the students and young people in general (like schematic playing games just to have a break), positive influence on their mentality (logic and strategic games) and, the most important, very clear negative (sometimes statistically proven) mental impact.

The urgent attention is necessary when the students unusually use to play the fighting or war games with the multiple lives option. Those games give, thanks to their very attractive 3D options, the feeling that the player remains in some fantastic virtual reality where he is to kill the enemies having guaranteed the next life in the case of some failure. Naturally, the students interested in such games try to cross the next and next levels trying to be as efficient as possible, to be faster and faster and, finally, to compete with each other. The more they play, the virtual reality becomes closer to the reality and, during longer sessions, it starts to interact and to mix with the real life (not only academic). It is observed during the exams and the projects evaluation that they start to treat the education process like the next game with the next life option, where some failure at any level does not really matter, since there is still a next option. The next step is noticed when the teacher is supposed to make a mistake when this teacher is not in a position to offer the next chance to have an evaluation. Therefore, the real life appears to be a weaker version of this game since no option to correct this mistake and, which is essentially more dangerous, that the errors cannot be simply cancelled from the overall memory by a simple resetting of this bad situation. This player mentality with a permanent interference of the virtual and the real worlds is especially visible when the students having some errors in their projects bring once more the same projects without any corrections to the supervisor expecting that he will not notice them (taken from a personal experience of the author). Then, the next life and the game interacts with the social contacts with the supervisor — most probably, if he notices that he will have the next option (my next life from the game) to correct them then everything (like after the special button pressing) remains the same as before. As it is widely known, the key moment of the engineering studies is in the projects preparation, mainly using the computers, where this type of a mental behavior leads to the disaster that any error may be corrected and has no real influence on the final result. The project is also a kind of a game, because it is carried out on the same virtual platform, so that any mistake is not really important since there is the next life. It is not necessary to explain the engineers how dangerous it can be in real designing process, where a single mistake of the designer can cause a total disaster and cannot be avoided on a certain manufacturing phase. It is hard to imagine a population of the people having academic education and thinking in the way that all humans' activities are the subject of a game with the total renew options always active. There is a single, but important, exception from this rule and these are the fight simulators in military schools, where the students must be prepared to fight and to train the war games with the enemies like terrorists.

Concluding those facts and observations, we must say that enormous interest in the games with multiple lives options is dangerous for the students, must be prohibited during the education of computer science. Furthermore, having in the mind that the computer games are so popular and they are not bad as a rule, it is advised to spend the few minutes to recommend the students the most thoughtful (and attractive at the same time) computer games and to explain the students a motivation to his choice. It may positively influence the lecturer position. As the example may serve the game called HEX (easy available as the freeware HEXY) invented by American Noble prize recipient John Nash, which received the gold medal on the computer Olympic Games in Athens in 2000. It has a big chance to be a popular competition because a history of John Nash became popularized last years by the separate cinema production. The game is somewhat similar to the historical game GO, but the dimensions of the rhomboidal game plane can be modified by the player, who, having the stones with the specific color, needs to create a path form one side of his color to the other (move by move with the computer as the second player).

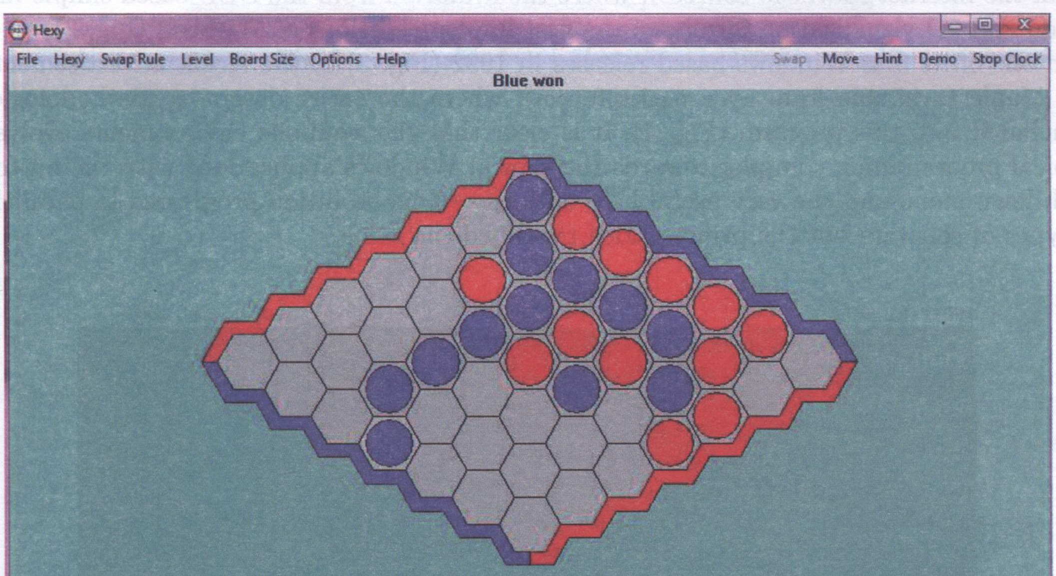

Fig. 1. Hexy screen

3. THE GENERAL NOTES ON SYMBOLIC SOFTWARE AND ITS EVOLUTION

According to various definitions, the symbolic computing is a discipline focused on the representation and a manipulation of the information in the form of a symbol, which is mainly connected with its numeric representation for further processing. It is programmed to automate complicated algebraic manipulation with such equations, also differential and integro-differential equations. The symbolic computer programs offers a large variety of the mathematical internal functions and procedures together with graphical options providing a programming language to define some users procedures. From the technical point of view, the symbolic software is implemented as the directed, aperiodic graphs but this programming algorithm and the code remain still unavailable for the users at all.

A history of the symbolic software development starts at 1960's, thanks to the pioneering work of the Nobel Prize recipient Martin Veltman who programmed a prototype of this software known as *Schoonship*. The next, more professional systems were developed at the MIT at 1970's and early 1980's; the first most popular systems were known as muMATH, Reduce, Macsyma. The half of eighties brought the commercialization of the symbolic packages and the next, still popular even today implementations like Maple, Mathcad, Mathematica or Maxima. The migration of the symbolic software to scientific calculators was also noticed at this time like firstly in the case of Hewlett-Packard with its HP-28 having arrangements of the algebraic expressions, differentiation, limited symbolic integration, Taylor series expansion and the entire solver for algebraic equations system. Later on, Texas Instruments programmed its calculator TI-92 with the use of Derive package. Nowadays, the symbolic software is available in a plenty of realizations for different operating systems (Linux, all Windows versions, MacOS, for instance), for grid computing (like gridMathematica, v. 6.0, since 2007), in some national versions (see Polish Derive since 2005). A commercialization level and the market for those programs may be well illustrated by the fact that Mathsoft programming the system Mathcad has been acquired by Parametric Technology Corporation for 62,0 million USD in April of 2006. The progress of the symbolic software is visible in the three next slides (first two from the internet Mathcad sources), where the evolution from the DOS-based simple functions operation and 2D visualization for one of the first implementations of Mathcad (Fig. 2), we browse through one of its first Windows-based releases in 1992 (Fig. 3) to the recent advertisement of the system Maple (available from www.maplesoft.com), where the entire designing methodologies have been included into this program (Fig. 4). It is clear that the symbolic environments evolved from the typical programming compiler towards the typical Windows applications with the buttons, the scrolled down menu and the very useful help options — this evolution progressed in parallel to the other types of software but the primary idea remained the same.

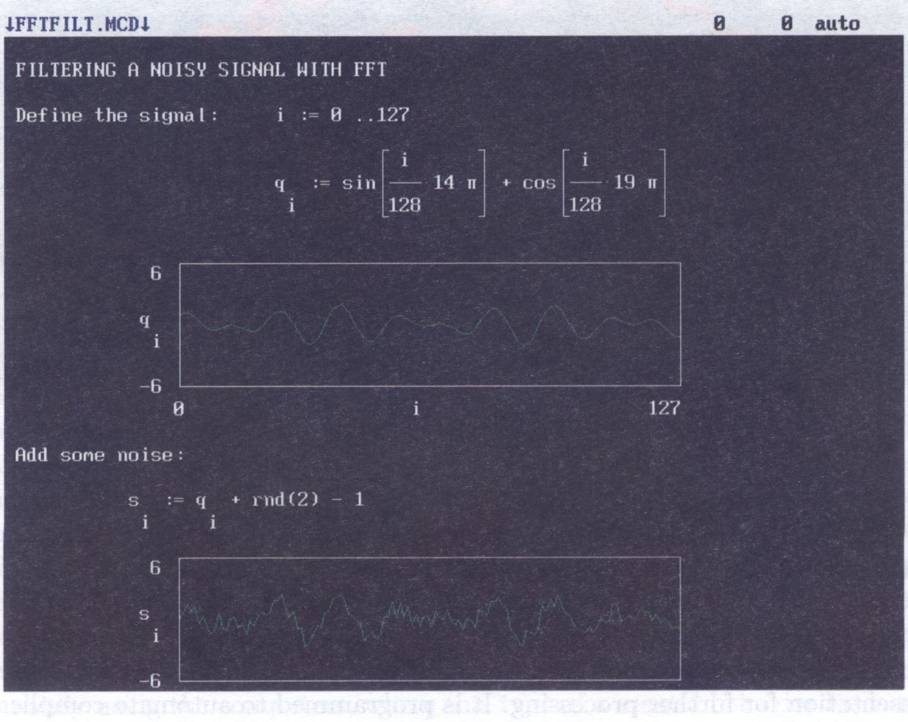

Fig. 2. One of the first releases of Mathcad

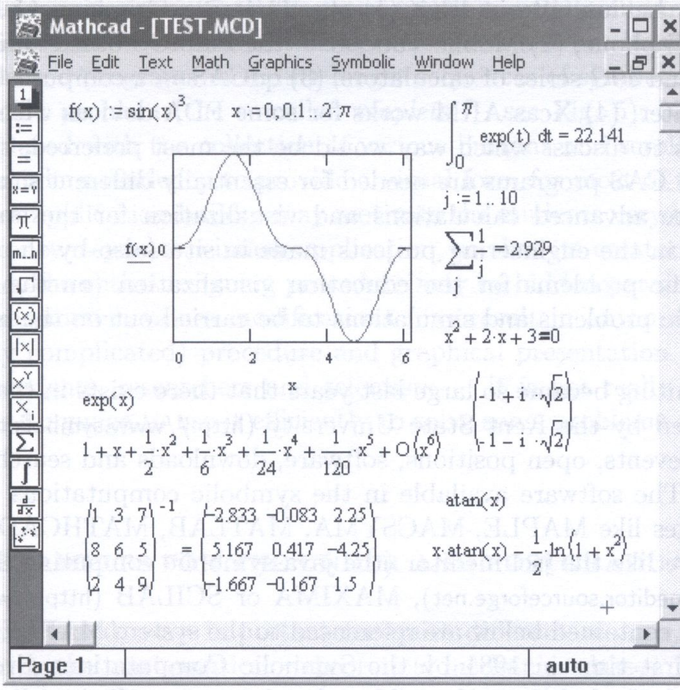

Fig. 3. The main window of Mathcad in 1992

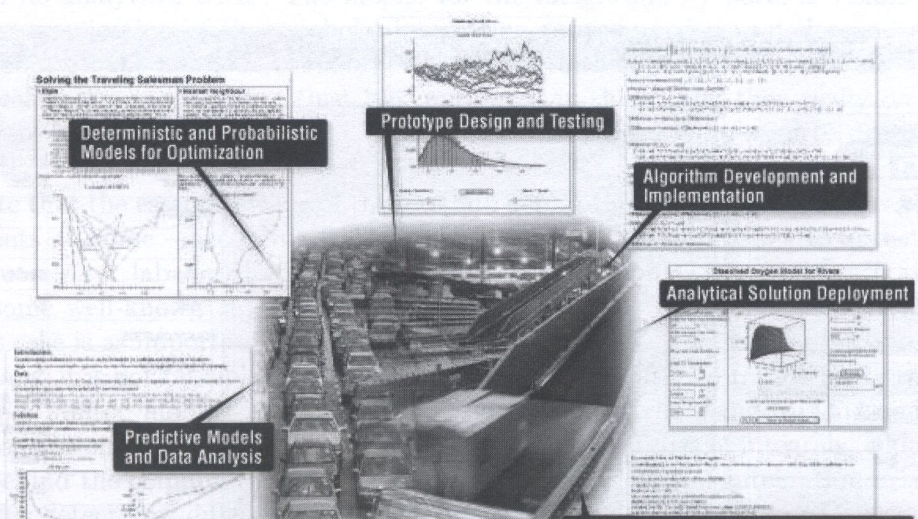

Fig. 4. Maple capabilities for its latest version 12.0 available since 2008

The future of symbolic computing is not always in personal and massive computers but also, maybe mainly, in popular portable devices implementations like mobile phones, modern scientific calculators and PDAs. Let us note at least (1) the TVH-72g Graphing Calculator works for the Sony Ericsson J300i cell phone, (2) Mobile Voodoo — the PalmOS based emulator for the Hewlett-Packard 48SX, 48GX and 49G series of calculators, (3) qdCAS — a computer algebra system for the pocket personal computer, (4) Xcas ARM works for some PDA devices with Linux and Windows. It is completely useless to discuss which way would be the most preferred, since various hardware implementations of the CAS programs are needed for essentially different scientific and engineering problems — from more advanced calculations and visualization for the students during classes, through some changes in the engineering projects made in situ (also by the portable devices) and the elementary scientific problems for the education visualization (on the classical PCs) to the real large scale scientific problems and simulations to be carried out on the multiprocessor massive computers.

The symbolic computing became so large last years that there exists in the internet the so-called SymßolicNet maintained by the Kent State University (http://www.symbolicnet.org) containing the conferences and other events, open positions, software, downloads and search engine as well as the subscription sub-site. The software available in the symbolic computations area obey a plenty of the commercial packages like MAPLE, MACSYMA, MATLAB, MATHCAD or MATHEMATICA as well as even freeware like the jscl-meditor (the java symbolic computing library and mathematical editor, http://jscl-meditor.sourceforge.net), MAXIMA or SCILAB (http://www.scilab.org). All the detailed considerations contained below are referenced to the system MAPLE (present version 12.0) implemented for the first time in 1981 by the Symbolic Computations Group at the University of Waterloo in Canada. Nevertheless, there is no loss of a generality in discussing everything on MAPLE and all the options recalled here do exist in the remaining systems and are easy available in the remaining systems after small semantic modifications only.

Let us remind now that the genesis of all those programs is more less the same — they have been initiated at the applied mathematics and/or computer science departments of the worldwide leading universities and through those years went out outside academic area by commercialization or are

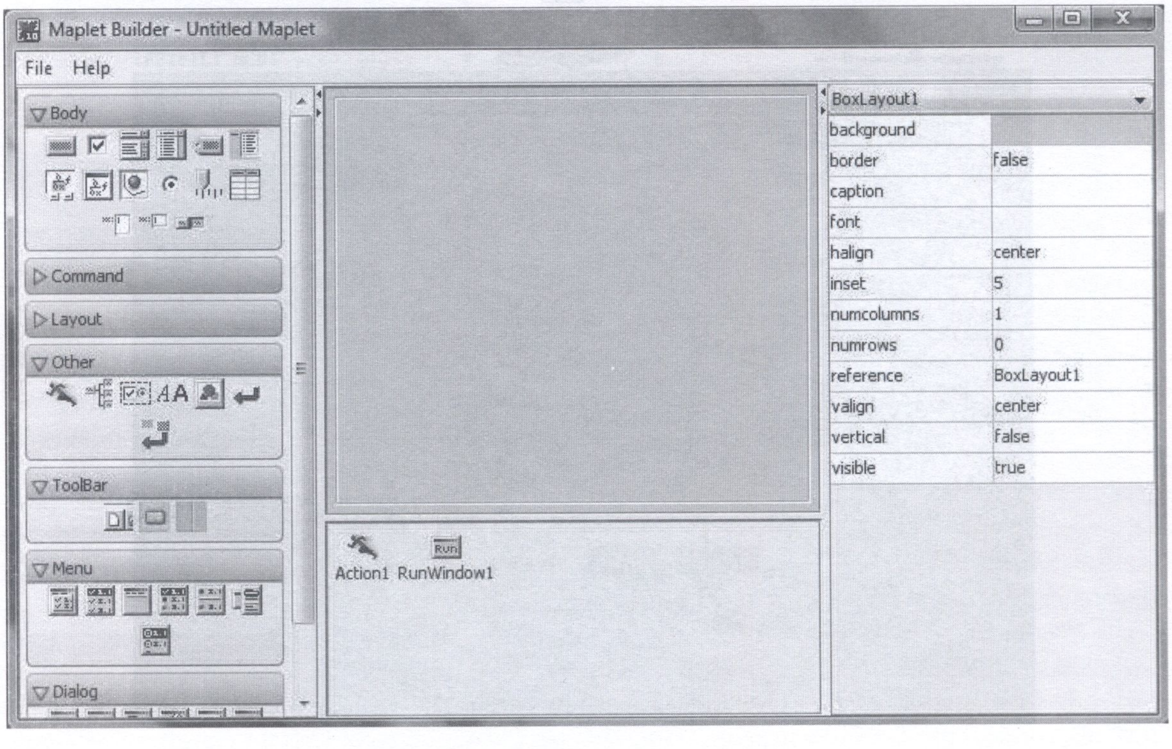

Fig. 5. The maplet builder window

still being developed by the scientists or engineers in many countries under the GPLs. Nowadays, the symbolic programs are very powerful and, besides all simple algebraic transformations and differentiation or integration techniques, have the entire ODEs and PDEs libraries, extended statistical as well as fantastic visualization and animation options. Last but not least, it is possible to build a separate window-released programs including longer sequences of symbolic operations devoted to a solution of the specific problem (the so-called *maplets* — see Fig. 5). Now, sometimes complicated semantics is replaced with the collection of various buttons and scrolled down menus, which perfectly reflects the modern tendency to provide a visual form for most of the necessary computer operations. The maplet application builder may motivate the students very much, because it is very attractive itself as the new separate window application, so that its creator is a good programmer and, on the other hand, the entire designing procedure may be hidden parametrically under a single window so that the parameters values modifications may result in automatic recalculation of the entire (even extremely complicated) procedure and graphical presentation. It gives the quite new opportunity for the students, researchers and scientists — it is not really necessary to know the MAPLE programming language to use it efficiently to solve some problems and to make the results visualization.

4. Educational examples for the under and postgraduates

Now, let us show the selected, very important for quite different reasons, MAPLE applications, which can be used during the mathematics or theoretical mechanics classes. This short review starts from the integration methods illustrations by the additional *maplet* shown below. It is obvious that the mathematics hidden under the icons is indeed very dangerous educational direction since our students become totally computer-dependent mathematicians and without the aid of a machine they can do nothing of course. A paradox of this phenomenon is that the symbolic programs first reduced mathematics to the commands and then, the symbolic programmers implemented the windows-based applications builders to create the educational applications demonstrating various mathematical methods and its analytical tricks. The *maplet* for the integration by parts is visible below — see Fig. 6; a pure application of the symbolic integration procedure does not demand of course any knowledge about all the methods available. Then, the students taught this operation without any methodological background lecture (just ‘int’ option) may think (and they really do so) that this procedure is so easy that this is the real ‘piece of cake’ and there is no need to deal with it longer than a minute.

Let us note that the essential value of this method is that the integration can be repeated reliably by the students at home – according to the time limitation it is impossible to repeat this process even in the computer lab. So that, training this process at home, the students may discover by themselves some well-known rules, resulting in some primarily primitive mathematical intuition, which in this case is as important as the fundamental knowledge. Contrary to the opinions of many mathematicians, it is not possible at all to obtain the same educational results using the blackboard only, because a limited number of the problems that can be solved using this methodology do not lead to the conclusions that there are quite clearly recognizable types of integrals, which, after some time spent behind the computer appear to be trivial. Since we do this integration in a step by step fashion, we can detect how different methods interact during the integration of the product of some basic functions. The students may easily change the integrated functions by themselves and train this integration process like a game.

Next, we have the example how to easily calculate the surface (as well as their volume next) for the user-specified surface of the revolution, which is integrated with the extended graphical options. All may understand rather easily from this window-based internal application within the MAPLE system the main general concept behind this calculation (see Fig. 7).

Contrary to the previous example, we train here the integration over the interval, so that we can modify this interval directly, we can make a visualization of this integral rotated over a specific axis,

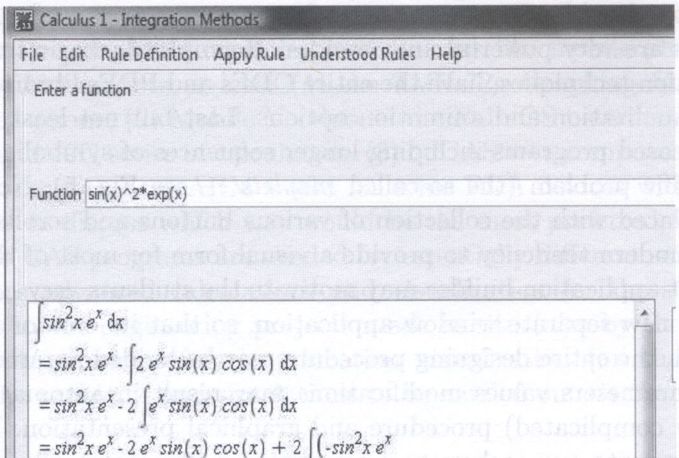

Fig. 6. The result of an integration in MAPLE

we may modify the initial function and the line of revolution as well as the distance of a rotation line to the specified coordinate axis (horizontal or vertical). The integral value appears in this window shortly after the integral to be computed on the basis of the user defined number of the partitions in the Riemann sum. So that, we can demonstrate that by increasing the number of partitions we obtain the convergence to the specific limit. Let us underline that the blackboard methodology can be based on rewriting of the basic formula and the computation of the single or two examples. It is not possible at all to make even similar figure on the blackboard, whereas the entire explanation needs a separate classes, so that having this into a single user-modified window we can expect from the students more understanding of this particular problem than before.

The next application of the MAPLE — to the vibrating mechanical system is shown here, where the solution is rather easy and the well-known (Fig. 8, with the amplitudes given parametrically in Fig. 9). The parameters of this system defined as follows: m[M]=1000 kg, m[U]=500 kg, d=100 000 N/m, whereas the basic solution of the differential equation itself reduces to the following three lines only:

restart; with(DEtools);
sol := dsolve(deq,x(t));
deq := (m[M] + m[U]) · diff(x(t),t,t) + c · diff(x(t),t) + d · x(t)
+ m[U] · r · ω^2 *· cos(omega · t) = 0;*

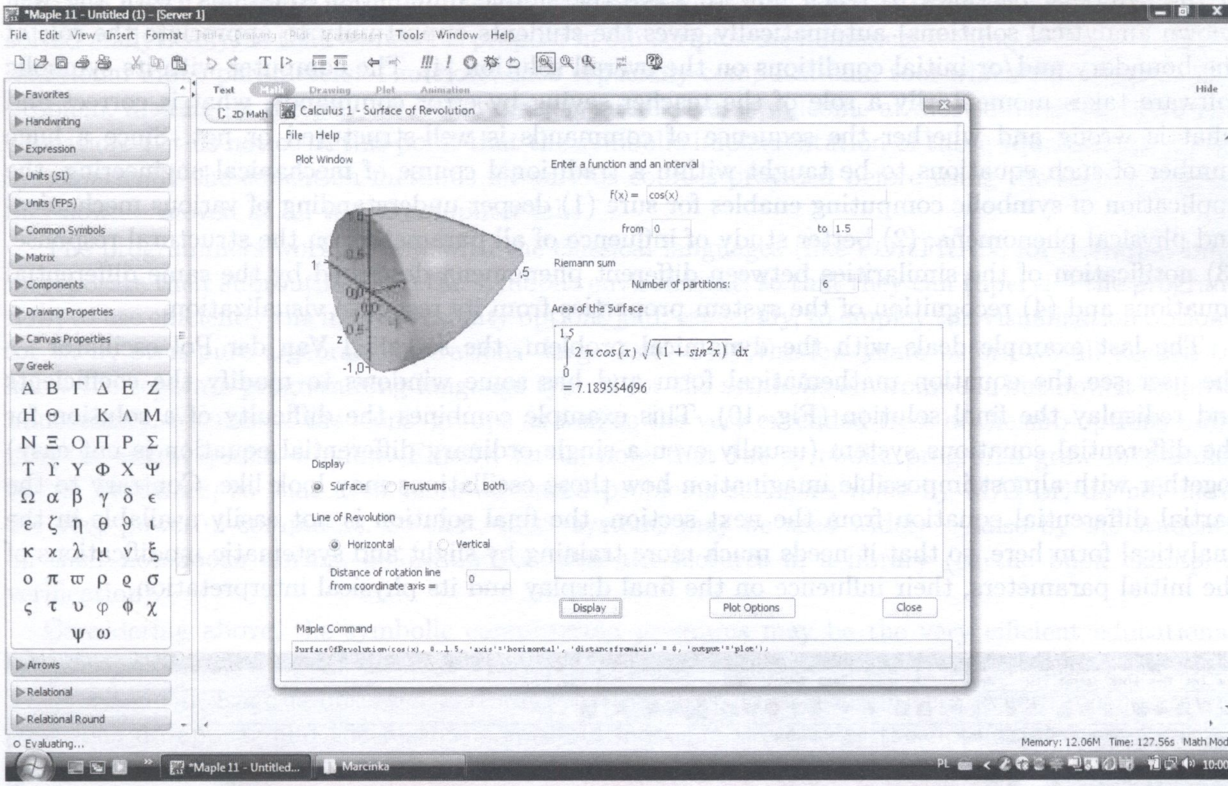

Fig. 7. The surfaces of the revolution modeling

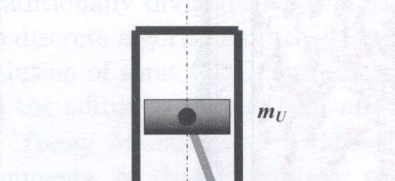

Fig. 8. The mechanical vibrating system

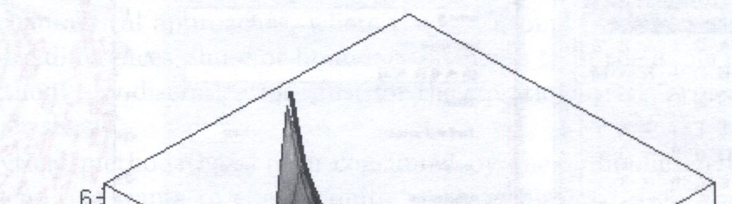

Fig. 9. Parametric display of the amplitudes for the vibrating system

This example is so easy that no *maplet* is necessary to obtain the solution here. An application of the symbolic packages to teach how to solve the simple differential equations (with the well known analytical solutions) automatically gives the students opportunity to appreciate the role of the boundary and/or initial conditions on the overall solution [4]. The computer with its symbolic software takes momentarily a role of the teacher saying by error commands, what is correct and what is wrong and whether the sequence of commands is well-structured or not. Since a huge number of such equations to be taught within a traditional course of mechanical engineering, the application of symbolic computing enables for sure (1) deeper understanding of various mechanical and physical phenomena, (2) better study of influence of all parameters on the structural response, (3) notification of the similarities between different phenomena described by the same differential equations and (4) recognition of the system properties from its response visualization.

The last example deals with the dynamical problem, the so-called Van der Pol oscillator — the user see the equation mathematical form and has some windows to modify the coefficients and redisplay the final solution (Fig. 10). This example combines the difficulty of a solution for the differential equations system (usually even a single ordinary differential equation is not easy) together with almost impossible imagination how those oscillations may look like. Contrary to the partial differential equation from the next section, the final solution is not easily available in the analytical form here, so that it needs much more training by slight and systematic modifications of the initial parameters, their influence on the final display and its physical interpretation.

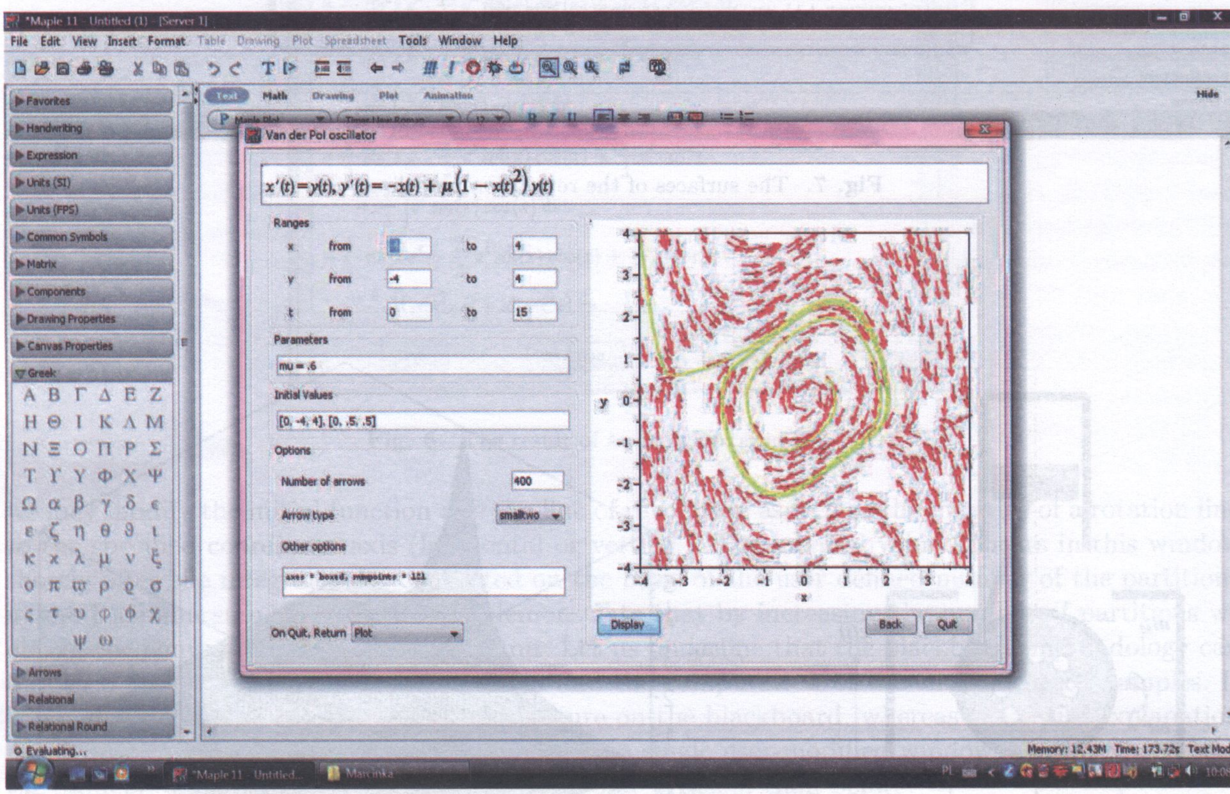

Fig. 10. The surfaces of the revolution modeling

This oscillator is just the illustration only how we can use the system MAPLE to make a solution and its visualization for different more complicated ODEs, PDEs and their systems appearing in different models and algorithms like theory of electronic circuits, various coupled problems in physics or so. It is not a secret that the solution to some of them need deeper mathematical knowledge and the students after multiple trials with no success, even with properly implemented equation, may give up or/and to fall into the conclusion that the MAPLE cannot solve this problem. Taking into account this difficulty, the software providers created some special automatic adviser to recognize the equation and its solution method. The separate problem is that some of those equations cannot be solved analytically, so that we need to propose in our script some numerical technique demanding the additional input parameters. Therefore, each especially computational tool which can simplify this problem, making by the way its reanalysis very attractive, is welcome at most during the academic education. Let us notice at this point that the symbolic implementation of those equation influences systematically the education methods for various courses provided before using the regular classes and now — moved at all to the computer labs.

The programmers working now with the classical languages (like FORTRAN, for example) may incorporate their subroutines into the symbolic environment, so that they can supervise the program flow, to use efficiently the interoperability options and, especially, to employ the visualization options for their, even pure algebraic, operations' large lists. Only the few years before we all needed to know the separate programming language typical for the symbolic environment but now it is quite unnecessary, because of the icons' groups as well as the very extended trees of the sub-options built into the main system window. Finally, let us note that the symbolic programs grow in parallel to the hardware, so that even more advanced problems solutions done by MAPLE do not need the very powerful computers. So that, those systems may be used widely — also by the students on their notebooks during, before and/or after the lectures in a library (as the book examples verification).

Considering above, the symbolic computation programs may be the very efficient educational tools implemented both in the university computer labs and at home — for a distant learning. Their extended usage is observed and grows naturally in all those subjects, where mathematical equations do appear and the reason is two-fold here: (1) there exist the applications centers and the internet well-documented user's manuals, where you can find a lot of the ready-to-use problems solved before and commented sufficiently by their authors; (2) the user does not really need to know most of the mathematical methods leading to his particular solution. This user needs to use the proper command (or the commands sequence in a worse scenario), sometimes even the application name and then just click simply the ENTER button or the additional icon. The exponential growth of the users may be explained taking into account a qualitative progress of the numerical methods in the applied sciences. It is widely known, that almost the few years ago the methods available were traditionally divided into analytical and numerical approaches, where the last group was adequate to discrete algorithms rather like the finite differences, finite or boundary elements than the explicit solution of some ODEs system made without any discretization (just for the equations transformed to the editor as they appear on the paper sheet).

Today we can observe that all analytical methods have been consumed by the symbolic environments, so that for simpler problems no one wants to engage quite separate discretization and the computer program when the solution and visualization is easy in the CAS programs (and may be at once slightly modified to check the parameter sensitivity of an answer). On the other hand, the academic software for the scientific computations written in lower level languages may be easily incorporated here into the exciting graphical platform, so that we theoretically can have everything in a single computer program realized according to our philosophy. Since the leading computer software companies systematically develop their systems autonomously, our 'ugly' computational routines written long ago may get a year after year the new packaging and the interaction options automatically. The small group of the hybrid analytical-numerical methods including initially symbolic algebra may become in the future the largest class on the market including most of the computational software.

The main role of the symbolic programs is to enable the students to discover the existing knowledge in an easier and decisively more attractive way than before. The very important tools are (1) a programming language very close to the natural English (some other versions like German, French or Japanese also available), (2) the very extended and interactive 'Help' reacting for the words similar to some internal procedures with a large variety of the ready-to-use examples for copy-paste usage, (3) graphical environment — the users may change the domain of both 2D and

3D plots and animations, multiple functions or parametric families may be plotted for the same intervals of the independent variables. Therefore, the students may distinguish between different mathematical models for the same phenomenon or the same mathematical model but with slightly modified parameters.

However, the key feature of the symbolic programs lies undoubtedly in the multilevel structure of mathematical operations, the solution verification and dynamical definition of the types and names. It is visible at most for the differential equations and their systems, which are commonly used in some basic sciences as well as almost the entire engineering. They are at the same time too complicated to be remembered even by better students.

The symbolic computing programs give the brand new opportunity for our students — an opportunity to discover the new facts and the research findings they cannot find at a single or even at any available book. Using mathematical definitions of some quantities and some symbolic operations they can derive (with the teacher's aid) some more advanced properties of these quantities or the properties of their functions (like products, the powers or so). It needs essentially more mathematical background and intuition but an opportunity to discover something new is the very exciting by itself, so that the educational process is done by the way only (by the way for the students, of course). The aspect of a time spent on the education is also very important here — having such a computational tool there is no need to (1) derive slowly and systematically everything on the blackboard, (2) recommend the students the additional books in the libraries, (3) prepare the new and extend the existing presentations, which may be decisive considering the still decreasing number of the lecturing and contacting hours with the students everywhere.

Finally, let us note that the MAPLE and related systems offer today the opportunity to prepare the entire lectures and even books into them. We may use to this purpose the automatic renumbering of the equations appearing into the text as well as automatic saving in the *.html format (some examples can be downloaded from the applications centers — http://www.maplesoft.com). There is also the huge library of the mathematical symbols, Greek letters as well as the other graphical tools to prepare the excellent presentation.

5. The example for the PhD students

As the educational example for the symbolic computations during the PhD courses (related to probabilistic methods and/or reliability analysis) we propose the following transient heat flow problem governed by the following ordinary second order differential Fourier equation [2, 7],

$$\rho c T - \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) - g = 0; \qquad x \in \Omega; \quad \tau \in [0, \infty), \tag{1}$$

where $c = c(T)$ means the random heat capacity characterizing some domain Ω, $\rho = \rho(T)$ is the random density of the material contained in Ω, $k = k(T)$ stands for the random thermal conductivity, while $g = g(T)$ is the deterministic rate of heat generated per unit volume; the variables T and τ denote temperature field values and time, respectively. Although it is possible to solve this problem using the crude Monte-Carlo simulation technique, which can be easily implemented in MAPLE using the internal statistical procedures, we use the stochastic perturbation technique based on the expansion of all functions and parameters in this equation into the Taylor series with random coefficients. The educational goal is to show how we can solve some transient problem with random parameters using the method, where the randomness is inserted using in a quite deterministic manner using the random quantities probabilistic moments instead of the large statistical populations.

This equation should fulfill the boundary conditions of the $\partial\Omega$ being a continuous and sufficiently smooth contour bounding the Ω region. The boundary conditions proposed for Eq. (1) are deterministic and they are defined as follows:

1. temperature (essential) boundary conditions

$$T = \hat{T}; \qquad x \in \partial\Omega_T\,, \tag{2}$$

and for $\partial\Omega_q$ part of the total $\partial\Omega$;

2. heat flux (natural) boundary conditions

$$\frac{\partial T}{\partial n} = \hat{q}; \qquad x \in \partial\Omega_q\,, \tag{3}$$

where $\partial\Omega_T \cup \partial\Omega_q = \partial\Omega$ and $\partial\Omega_T \cap \partial\Omega_q = \{\emptyset\}$;

3. initial conditions

$$T_0 = T(x_i;0); \qquad x_i \in \Omega, \quad \tau = 0. \tag{4}$$

It is possible to discuss the influence of randomness for the initial and boundary conditions later on, using the same example, to demonstrate the students an essential difference between the process, where the solution is deterministically given as 0 and the situation, when its expected value equals 0 and higher moments do not vanish.

Let us suppose further that heat conductivity and volumetric heat capacity are uncorrelated Gaussian random fields defined uniquely by their two probabilistic moments as follows

$$E[k], \ E[\rho c] \quad \text{and} \quad Var(k), \ Var(\rho c), \tag{5}$$

which completes the probabilistic description of physical properties of the composite constituents. This assumption is not so natural for this moment as for the statistical analysis, where it is necessary to generate numerically the distributions for those parameters. We will provide a symbolic solution to this problem in the form of the expected values and standard deviations of temperature field for this heated structure. Therefore, let us denote the corresponding random vector of the problem by $b(x)$, with probability density functions $p(b)$, so that the mth order central probabilistic moment is given by

$$\mu_m(b) = \int_{-\infty}^{+\infty} (b_r - E[b_r])^m \, p(b)\, \mathrm{d}b. \tag{6}$$

As it is known, the basic idea of the stochastic perturbation approach follows the classical perturbation expansion idea and is based on approximation of all input variables and the state functions of the problem via truncated Taylor series about their spatial expectations in terms of a parameter $\varepsilon > 0$. For example, in the case of random heat conductivity, the nth order truncated expansion may be written as [6]

$$k = k^0 + \varepsilon k^{,b} \Delta b + \frac{1}{2}\varepsilon^2 k^{,bb} \Delta b \Delta b + \cdots + \frac{1}{n!}\varepsilon^n \frac{\partial^n k}{\partial b^n}(\Delta b)^n \tag{7}$$

where

$$\varepsilon \Delta b = \varepsilon (b - b^0) \tag{8}$$

is the first variation of b about its expected value and, similarly,

$$\varepsilon^2 \Delta b \Delta b = \varepsilon^2 (b - b^0)^2 \tag{9}$$

is the second variation of b about its expected value, where nth order variation can be expressed accordingly. We can use the internal function of MAPLE to provide this expansion automatically. The temperature may be expanded accordingly as

$$T = T^0 + \varepsilon T^{,b}\Delta b + \frac{1}{2}\varepsilon^2 T^{,bb}(\Delta b)^2 + \cdots + \frac{1}{n!}\varepsilon^n \frac{\partial^n T}{\partial b^n}(\Delta b)^n. \tag{10}$$

Traditionally, the stochastic perturbation approach for all the physical problems is entered by the respective perturbed equations of the 0^{th}, 1^{st} and successively higher orders being a modification of the variational integral formulation. Hence, there holds

1. one zeroth-order partial differential equation

$$\int_\Omega \left(\rho^0 c^0 \dot{T}^0\, \delta T + k^0 \frac{\partial T^0}{\partial x}\, \delta\frac{\partial T}{\partial x}\right) \mathrm{d}\Omega = \int_{\partial\Omega_q} \hat{q}^0\, \delta T\, \mathrm{d}(\partial\Omega) + \int_\Omega g^0\, \delta T\, \mathrm{d}\Omega, \tag{11}$$

2. nth order higher order partial differential equation

$$\int_\Omega \delta T \left(\sum_{p=0}^{n}\binom{n}{p}\rho^{(p)}\sum_{m=0}^{n-p}\binom{n-p}{m}c^{(m)}\dot{T}^{(n-p-m)}\right)\mathrm{d}\Omega$$

$$+\int_\Omega \delta\frac{\partial T}{\partial x}\sum_{p=0}^{n}\binom{n}{p}k^{(p)}\frac{\partial T^{(n-p)}}{\partial x}\,\mathrm{d}\Omega = \int_\Omega \delta T(g)^{(n)}\,\mathrm{d}\Omega + \int_{\partial\Omega_q}\delta T(\hat{q})^{(n)}\,\mathrm{d}(\partial\Omega). \tag{12}$$

The symbolic computations program may be very useful at this moment also for automatic derivation of the increasing order equations (formation of those equations and collection of the same order components). Having solved those equations for T^0 and their higher orders respectively, we derive the expressions for the expected values and the other moments of the temperature field. In order to calculate the expected values and higher order probabilistic moments of temperature $T(b;t)$, the same Taylor expansion is employed to the definitions of probabilistic moments

$$E[T(t,b);b] = \int_{-\infty}^{+\infty} T(b)\,p(b)\,\mathrm{d}b = \int_{-\infty}^{+\infty}\left(T^0 + \varepsilon T^{,b}\Delta b + \cdots + \frac{1}{n!}\varepsilon^n T^{,n}(\Delta b)^n\right)p(b)\,\mathrm{d}b. \tag{13}$$

Since the symbolic integration is available in MAPLE, we can do that in two different ways — by a direct integration according to this formula, and, on the other hand, by employing this perturbation under the integral and using the definitions holding true for Gaussian variables (for the standard deviations σ)

$$\mu_{2k+1}(b) = 0, \qquad \mu_{2k}(b) = (2k-1)!\,\sigma^{2k}(b); \qquad k \le m. \tag{14}$$

Using this extension of the random output, a desired efficiency of the expected values can be achieved by the appropriate choice of m and ε corresponding to the input probability density function (PDF) type, probabilistic moment interrelations, acceptable error of the computations, etc.; this choice can be made by comparative studies with Monte-Carlo simulations or theoretical results obtained by direct (i.e. symbolic) integration. Once more the same symbolic computations package seems to be the best computational tool because we can operate the additional polynomial forms directly; the same general implementation in some programming language could be extremely difficult and could be the problem itself.

The computational part is provided using the following data: $E[k] = 1.0\,\mathrm{BTU}/(\sec\,\mathrm{in}\,^\circ\mathrm{F})$, $E[\rho c] = 1.0\,\mathrm{BTU}/(\mathrm{in}^3\,^\circ\mathrm{F})$, where the deterministic result [2] is provided using those expectations instead of their deterministic counterparts

$$T(x,t) = \frac{2}{k}\left\{\left(\frac{\kappa t}{\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{x^2}{4\kappa t}\right) - \frac{x}{2}\, erfc\left(\frac{x}{2\sqrt{\kappa t}}\right)\right\}, \tag{15}$$

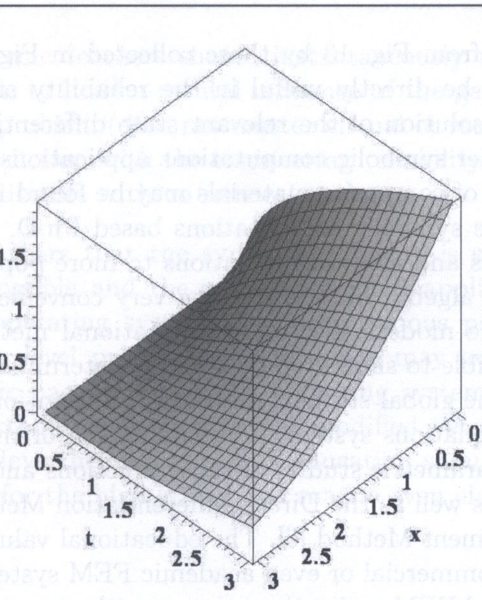

Fig. 11. Temperature distribution along the heated rod in time

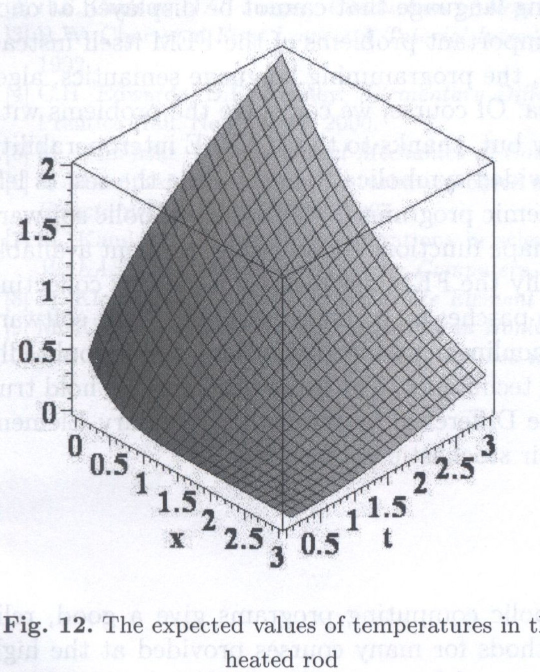

Fig. 12. The expected values of temperatures in the heated rod

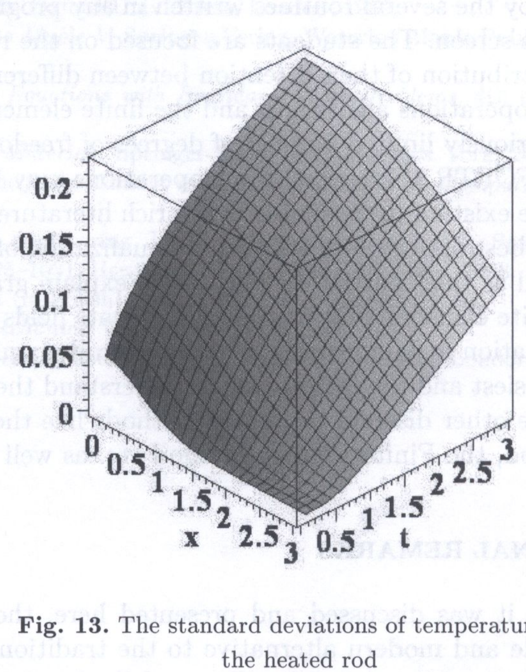

Fig. 13. The standard deviations of temperatures in the heated rod

where $\kappa = \frac{k}{\rho c} = 1.0\,\frac{\text{in}^2}{\text{sec}}$ is further used for the probabilistic modeling. Thanks to the application of the symbolic computations package it is possible to visualize the deterministic solution (Fig. 11) with respect to the spatial and temporal coordinates at the same time, which is obtained automatically at once. We may point out here that this particular heating process, against the homogeneous character of the rod, has essentially different character at the opposite ends of this structure — let us compare at least $x = 0$ with $x = 3$. It is also possible to make an animation showing dynamical temperature changes within the heated rod as well as to carry out the parametric studies by a collection of different surfaces on a single graph resulting from different combinations of heat conductivity and volumetric heat capacity.

Furthermore, we investigate the probabilistic problem, where the expected values and standard deviations are visualized. As it is clear from Figs. 12 and 13 reporting those computations, the expected values during the entire heating process are larger than the corresponding deterministic values; the coefficients of variation of the temperature are equal more less 0.1, which can be recovered dividing the results from Fig. 13 by those collected in Fig. 12, respectively. This study after small modifications may be directly useful in the reliability studies of such structures and for all those cases, where the solution of the relevant state differential equation may be symbolically obtained. A plenty of other symbolic computations applications in homogenization, strength, plasticity, fracture and fatigue of composite materials may be found in [5].

Finally, let us note that the symbolic computations based Ph.D. courses may be provided not only for the analytical methods and analytical solutions to more popular ordinary and partial differential equations. The linear algebra library give the very convenient tools for an illustration of the entire discretization by the modern discrete computational methods like the Finite Element Method for instance. It is possible to show step by step (a) determination of the elemental stiffness matrices, (b) composition of the global stiffness matrix, (c) imposition of the boundary conditions, (d) solution of the algebraic equations system, (e) visualization of the final displacements, strains and stresses, (f) provide the parametric studies of those functions and (g) carry out the sensitivity studies using both analytical as well as the Direct Differentiation Method [9] or analogously implement the Stochastic Finite Element Method [6]. The educational value of the symbolic programs is significantly higher than the commercial or even academic FEM systems because (a) all operations are visible (unlike in commercial FEM realizations that must be treated as the black box modelers), (b) it is easier to explain the crucial moments and tricks of the FEM on the single MAPLE script than by the several routines written in any programming language that cannot be displayed at once on the screen. The students are focused on the really important problems of the FEM itself instead of distribution of their attention between different files, the programming language semantics, algebraic operations algorithms and the finite elements idea. Of course, we can solve the problems with the seriously limited number of degrees of freedom only but, thanks to the MAPLE interoperability with FORTRAN sources, some operations may be provided symbolically here, while the rest is left for the existing (or available in the rich literature) academic programs [8, 10]. The symbolic software is the best opportunity to make a visualization of the shape functions for any finite element available as well as their partial derivatives, to explain graphically the FEM discretization idea by collecting all finite elements and their displacements fields as the patches on a single graph etc. This software application in demonstration of the elastodynamics or nonlinear problems solutions are undoubtedly the easiest and the fastest way to understand the FEM techniques. The same observations hold true for the other discrete computer methods like the Finite Difference Method, the Boundary Element Method, the Finite Volume Method etc. as well as their stochastic extensions.

6. Final remarks

1. As it was discussed and presented here, the symbolic computing programs give a good, reliable and modern alternative to the traditional methods for many courses provided at the high schools and universities. Successfully for the teachers and lecturers, they still evolve to enlarge an educational impact on the students and to fasten the computational part of a given problem solution. Now, the symbolic programs remain educationally extremely worthy because, thanks to their visual attractiveness comparable to many games and the internal language very close to the natural (English standard), they impose an algorithmic thinking of the users and, on the other hand, an opportunity to implement many important problems from the basic sciences and almost all engineering problems. According to the time limitations for all computer science lab classes and capabilities of the CAS systems large enough for several academic courses, the examples solved for and by the students must be well prepared and specifically chosen to find the most crucial features, options and the solution methods.
2. It is important, as it was discussed above, that a lecturer of the computer science should be a supervisor of the students not only for the classes but also of their general activity on the computers – to avoid the usage of the unverified and improper knowledge sources existing in the internet and created sometimes by the students to speed up their educational successes. On

the other hand, the academic teachers should instantaneously monitor playing games by the students to prevent or even to limit a mental influence of the games with multiple lives, where they are taught that each problem (does not matter virtual or real) has several solution trials. It is in a fundamental contradiction with the engineering reliability concept, where each structure or system has a single service life with no renewal options.

3. It becomes apparent last years that the symbolic software is so popular, it has so universal character and so many possible and the well-documented applications that it appears on all types of the computers, operating systems and even various portable devices. Its popularity resulting from the highest level programming language may even lead in some future to an integration of the symbolic packages with the operating systems, so that the new computers will be equipped (like portable devices) with the modified versions of the software belonging today to the CAS class. Nevertheless, its role in education should instantaneously increases in all academic courses, also for the high school and maybe even elementary courses.

References

[1] I.N. Bronstein *et al. Taschenbuch der Mathematik* (in Polish). Polish Sci. Publ., Warsaw, 2004.
[2] H.S. Carlsaw, J.C. Jaeger. *Conduction of Heat in Solids.* Oxford Univ. Press, London, 1959.
[3] B.W. Char *et al. First Leaves: A Tutorial Introduction to Maple V.* Springer-Verlag, Waterloo Maple Publishing, 1992.
[4] C.H. Edwards, D.E. Penney. *Elementary Differential Equations with Boundary Value Problems*, 4th edition. Prentice Hall, New Jersey, 2000.
[5] M. Kamiński. *Computational Mechanics of Composite Materials.* Springer-Verlag, London–New York, 2005.
[6] M. Kamiński. Generalized perturbation-based stochastic finite element method in elastostatics. *Computers and Structures*, **85**(10): 586–594, 2007.
[7] M. Kamiński. Symbolic computations in science and engineering. In: E. Mastorakis *et al.*, eds., *Proc. 12th WSEAS CSCC Multiconference on Computers*, pp. 1025–1031, Heraklion, Greece, WSEAS Press, 2008.
[8] M. Kleiber. *Introduction to the Finite Element Method* (in Polish). Polish Sci. Publ., Warsaw, 1986.
[9] M. Kleiber *et al. Parameter Sensitivity in Nonlinear Mechanics.* Wiley, New York, 1997.
[10] O.C. Zienkiewicz. *The Finite Element Method in Engineering Science*, 2nd edition. McGraw–Hill, London, 1971.