AC 2009-334: TEACHING PHYSICS WITH COMPUTER ALGEBRA SYSTEMS

Radian Belu, Drexel University

Alexandru Belu, Case Western Reserve University

Teaching Physics with Computer Algebra Systems

Abstract

This paper describes some of the merits of using algebra systems in teaching physics courses. Various applications of computer algebra systems to the teaching of physics are given. Physicists started to apply symbolic computation since their appearance and, hence indirectly promoted the development of computer algebra in its contemporary form. It is therefore fitting that physics is once again at the forefront of a new and exciting development: the use of computer algebra in teaching and learning processes. Computer algebra systems provide the ability to manipulate, using a computer, expressions which are symbolic, algebraic and not limited to numerical evaluation. Computer algebra systems can perform many of the mathematical techniques which are part and parcel of a traditional physics course. The successful use of the computer algebra systems does not imply that the mathematical skills are no longer at a premium: such skills are as important as ever. However, computer algebra systems may remove the need for those poorly understood mathematical techniques which are practiced and taught simply because they serve as useful tools. The conceptual and reasoning difficulties that many students have in introductory and advanced physics courses is well-documented by the physics education community about. Those not stemming from students' failure to replace Aristotelean preconceptions with Newtonian ideas often stem from difficulties they have in connecting physical concepts and situations with relevant mathematical formalisms and representations, for example, graphical representations. In this context, a computer algebra system provides a better tool which is both powerful and easy to use. Their appropriate use can therefore be an important aid in the training of better physicists and engineers. In this presentation we will discuss ways in which computer algebra systems like Maple, Mathcad, Macsyma or Mathematica can be used, by instructors and by students, to help students make these connections and to use them once they are made. Benefits that accrue to upper-class students able to make effective use of a computer algebra systems provide a further rationale for introducing student use of these systems into our courses for those who plan to major in physics or other technical fields.

1. Introduction

Physics is guided by simple principles, but for many topics the physics tends to be obscured in the profusion of mathematics. As interactive software for computer algebra, such as Maple, MathCAD, Mathematica or MATLAB can assist educators and students to overcome the obstacle of mathematical difficulties or to improve the lecture presentations via power visualization, animation and graphic facilities of these software packages. The educators and students can take the advantages of the mathematical power of symbolic computation so they can concentrate on applying principles of setting equations, instead of technical details of solving problems. Moreover, most undergraduate physics textbooks were written before advanced computer algebra software became conventionally available. The conventional approach to a topic places emphasis on theory and formalism, devoting many paragraphs to performing algebraic or calculus operations in deriving equations manually, and other than some well known examples, most applications of theory are omitted. One reason that those examples are well known is that they admit analytic solution: they typically represent simplified solutions that generally fail to fully reflect the reality. In most situations, analytic solutions simply do not exist, and one cannot proceed without the assistance of a computer. Although some textbooks have sections discussing numerical methods, many of them contain just the theory of numerical methods, and one is required to posses programming skill for practice; this part is hence generally neglected. Essentially all experiments in physics measure numbers, so any formulation must eventually be reducible to numbers. Under a conventional curriculum, a student's ability to calculate and to extract numerical results from the formalism is somewhat inadequate. The result is not surprising: a student may be weak in those areas, and he or she thus achieves only partial comprehension because of technical difficulties. Computer algebra systems (CAS) can remedy some of these deficiencies or weaknesses in traditional education process and training. Using CAS, one can manipulate equations and diminish tedious paper work that distracts from main focus of learning physics. To become proficient problem-solvers, physics or engineering students need to form a coherent and flexible understanding of problem situations with which they are confronted. Still, many students have only limited representations of the problems on which they are working. In introductory physics courses a rich understanding of situations is more useful than procedural ability [1]. When students start to learn calculus-based physics the emphasis is shifted. Although situational understanding and the ability to identify a problem remain crucial to deep understanding and problem solving [2, 3], learning to carry out solution procedures simply consumes a large portion of the students' attention and takes up the available time. Therefore, it has been unavoidable that more challenges are postponed until procedural mastery has been achieved. Recent development in user-friendly computer algebra software may offer new opportunities and tools to do some more substantial analysis in calculus-based physics courses.

This paper discusses the use of Computer Algebra Systems (CAS) in physics education as a teaching and learning aid. A brief overview of the challenges and problems of computer algebra-based lecturing and learning is given. From this point of view, the power and limitations of CAS as systems for doing mathematics and simulations, calculators with infinite precision, teaching-tools for non-trivial examples, and learning-tools for experimental examples are shown. New skills are necessary in order for students to manipulate symbolic computation programming languages and to judge the results; the new skills are discussed and it is argued that the fear that students will forget their basic mathematical knowledge is unjustified. A system of learning and teaching support modules of various physics topics developed and/or underway to be developed by the authors are presented and discussed. We believe it is worthwhile to develop new ways of teaching and learning physics, by taking advantage of the unprecedented developments of the last two decades in computer hardware, software, programming languages and Internet. The materials presented herein can be used as the starting point for other instructors considering using similar tools in undergraduate level physics courses. The authors also strongly believe that discussions and feedback from other educators will advance physics education through introduction of new topics, laboratory experiments or new emerging computer applications in delivering lecture or in doing experiments, as well in the development of new courses, new methods in supporting teaching and learning physics and help of faculty, especially the younger ones interested in research and teaching in this field.

2. Computer Algebra Systems Features and Physics Applications

Computer algebra systems have from their earliest days been concerned with providing tools with which researchers and scientists in other fields can determine new results. A computer algebra system (CAS) in itself is no more than a high level programming language for visualization, symbolic and numerical computation. Basically, computer algebra systems are programs designed for symbolic manipulation of mathematical objects such as polynomials, vector and matrix manipulations, integrals, equations, etc. Typical actions are simplification or expansion of expressions, solving (systems of) differential or algebraic equations, data analysis and statistical methods, etc. Most CAS allow the user at least to write sequential programs for complex tasks, in a manner similar to writing mathematical equations, and have all features of high-level programming languages available. As well as such features, CAS also have most of the features of numerical systems for visualization (2-D plots, 3-D plots, animations) and numerical computations (numerical equation solving, numerical integration and differentiation). However, numerical systems are typically faster in regards to the numerical handling of floats with fixed precision. Some CAS packages solve these problems by offering links to such numerical software as MATLAB (i.e. MAPLE V). Besides being a tool for the manipulation of formulas, CAS should be expert systems knowing all of mathematics in a good mathematical handbook. This has not really been achieved yet, but significant progress was made in the last decades, and it is expected that a CAS should know all integrals found in, for example Gradshteyn and Ryzhik [4] and all differential equations from Coddington's book [5]. The first computer algebra systems, which become available in late 1980s, were mainly of only theoretical interests. Over the last two decades, some of these software packages have evolved into more practical computation and visualization tools that can take over many routine problem solving tasks. At the same time the required hardware has become more affordable.

Computer algebras was from the very begging a tool for building activities?, and was accepted without reservations by physicists and theoretical chemists from the earliest days of symbolic computation. One of the earliest areas of CAS applications in physics was that of celestial mechanics, as well classical mechanics where it becomes an everyday tool for many researchers. In many applications in this area, such problems as gyroscopia, space dynamics, obits' computation, or the representations of the equations of motion in symbolic form avoids unreasonably large numerical experiments and simulates effective usage and development of algorithms for qualitative methods of analysis of equations constructed. In these areas, CASs usually suggest substantial aid both in the modeling stage (construction of the kinetic energy and the force function for mechanical system, derivation of equations of motion) and during qualitative analysis of obtained equations. This aid is appreciable even for objects of moderate dimension. Another area where CAS was useful is general relativity, with applications such as classification of Riemann tensor based on studies of the multiplicity roots of a quartic equation or on the equivalence problem. Quantum theory and high energy physics have been other active areas for the applications of symbolic computation. A good example is the use of the algebra systems in quantum field theory to check the accuracy of the answer with experimental results. Electromagnetic field theory is one of the areas of physics and engine engineering where symbolic computation is applied on an extended scale due to their capabilities in solving differential equations and visualization and graphic capabilities.

Some of the advantages of using a CAS packages are: a) students can write down mathematics in a programming-like way, using symbolic notations; b) less time spent with calculations leaves more time for physical analysis; c) geometric visualization of results; d) learning and become proficient in a high-level programming language; and e) the availability of free software applications, using well-documented algorithms. Dirive and Mathcad are already implemented on a pocket calculator, and more extensive packages, such as Mathematica and Maple, run on any desktop computer. In several branches of mathematics, physics and engineering, computer algebra systems have seen increasing popularity as a tool for constructing proofs, solutions and visualizing the results. Also in introductory mathematics courses at the university level, there is an increasing use of computer algebra software packages in teaching and learning. However, there are fewer examples where computer algebra systems were integrated throughout physics courses, especially at the introductory levels. That is not to say that computers have not been used extensively in physics and engineering courses, but their use has been mainly restricted to numerical applications, course delivery, presentations, data analysis, simulations, which are central to a calculus-based course. This implied that the central part of the course – introducing the theory, and proving the formulas – had be done most of the time by hand, more or less in student assignments. In this study we will argue that a CAS could be used, via several examples to promote students' understanding of problems and to support the formulation of associations between problem representations and solution information and a didactic approach for using such software to improve learning and teaching process in physics will be suggested.

There are many commercial and non-commercial products available. The most popular are Mathematica™ [9] and Maple™ [10] which will, in a (hopefully) everlasting contest, continue to evolve. Other systems are REDUCE™ [13], AXIOM™ [11], MuPAD™ [12] or Derive. All systems can be used for high-school to university mathematics, but they differ in comfort and complexity and each has a different look and feel. There are also some so-called hybrid software packages that allow symbolic computations as a feature of numerical systems (Symbolic Toolbox for MATLAB™ [14], Mathcad™ [15], and PV Wave™ [20]), and text processors (Scientific Workplace™ [17]) that have embedded a full CAS. All these programs contain a kernel of the Maple CAS. The problem with such hybrids is that in general they are fixed to a certain release of the underlying kernel or linked CAS and that normally they could not be used across platforms. Throughout this paper, Maple V is used as exemplary CAS, for two reasons: first one of the author preference and the second its availability at our universities. However, for most of the points discussed here it is a simple matter of taste as to which programs are used. Computer Algebra Systems can have a significant impact on the way mathematic, physics or engineering courses are taught and applied. The situation can in some sense be compared to the pocket calculators. Today, even in primary or elementary schools, these are simply a tool and it has not meant to decline of mathematics. It is however no longer necessary to memorize the multiplication table up to twenty-five. In teaching mathematics or physics now, it is possible to concentrate on mathematical or physics content, rather than on counting numbers of finding solutions of the exotic equations or integrals. By using CAS it is possible to go one step further. Instead of training integration rules one exotic case over and over again, for example, it is possible to concentrate on the meaning of a physics problem and variants of it. We are also no longer limited to trivial examples that work. Students are invited to play with physics. They learn that real life examples normally do not lead to closed formulas. But they can even play with and visualize the results and different approximations and they also learn to judge the results. They also learn that there are a lot of mathematical tools, each with their own rights and applicability.

We attempt to devise an instructional approach to promote students' understanding of these problems and to support them in forming associations between problem features and solution methods. The approach is to use symbolic computation packages as tools for problem solving and visualization. A set of modules, such as: harmonic oscillator, electrostatics, etc. were implemented based on this instructional approach. Other models from the fields of thermodynamics, acoustics, electromagnetism, optics or quantum mechanics are underway to be implemented in the near future This approach in teaching physics is unconventional in several aspects: its content reflects needs for high-tech physicists and engineers, the approach is strongly computer-supported, symbolic computing and other IT tools are systematically applied, problem-solving skills are intensely stressed.

The primary purpose of traditional courses in physics and/or modern physics is to introduce the students to the concepts and ideas of the twenty-first century physics. The topics covered in these courses include usually dynamics, waves, heat and thermal physics, kinetic theory of gases, electricity and magnetism, fluid mechanics, acoustics, optics, special relativity, elementary quantum mechanics, and atomic, molecular, solid state, nuclear and particle physics.

3. The Learning and Teaching Process

Learning physics, in particular how to solve a given class of physics problems is a complex and time-consuming process. As a primer a student may listen to a lecture, read the appropriate physics textbooks, or interact with a computer simulation to become acquainted with the with the domain concepts. It is only after this first encounter, however, that the student begins to learn how to solve problems. The continued learning process first requires the learner to combine information from different sources, such as textbooks, physics problems' collections, previous problem-solving experiences, mathematics and physics pre-knowledge. Second, it requires the learner to go beyond the literal information presented in order to create understanding, to see implicit regularities, and to learn to routinely apply domain theories. It is common that impasses and misunderstandings arise during the process, and insight often comes only after a period of time and several attempts and trials. After the initial conceptual barriers have been overcome, it still requires considerable practice to become fluent in selecting and finding the right solution step in a particular circumstance in recovering from errors, and in carrying out the selected solution steps and in solving the specific problem. From information processing point of view, there are two relevant approaches to the learning process described earlier: one is the broad-class of production-rule theories; the other is the schema theoretic approach.

Current learning theories suggest that problem representations are best constructed by the students themselves, and that an adequate problem representation has to be constructed in context of real problem solving activity [2, 3]. Therefore, the approach used in this study was to support the formation of problem representations during practice problem solving aiming to make a proper situation analysis intrinsically rewarding, rather than having it imposed by a teacher. Our review of several learning supporting tools leads us to the conclusions that a computer algebra system may offer the right functionality to achieve this goal. Three properties of CASs are of importance: a) CASs demand precise specification of a problem, in a highly constrained formal specification language; b) CASs takes over algebraic calculations; and c) CAS packages have powerful visualization and graphic facilities. The required precise specification of the problem and the assistance in algebraic computations can be used to direct students' attention to the properties of the problem situation and/or to the theory and phenomena behind the problem.

Teaching physics with software for symbolic computation, as we pointed out in previous sections allows an instructor to explore a topic from several points of view: a formal statement in words, just according to the tradition, including emphasis on definitions of terms; an algebraic and symbolic treatment, which can expand to take advantage of the speed and scope of software for algebraic operations; numerical aspects, with test cases over a large range, with numerical examples used to introduce topics as much as practicable; graphics, showing geometrical interpretations in two or three dimensions, with animations, in a way that it is entirely new and impracticable using traditional teaching methods; focusing on phenomena rather than on methods on solving. The advantage of visualization can not be overestimated: a picture or a 3-D graphic representation of a phenomenon with no everyday life representation, such as an electromagnetic wave is certainly worth a thousand words of jargon, and makes the concept memorable to even a physics disinclined student. The capacity of contemporary software for symbolic computation to produce outstanding graphs and plots is astonishing; today teaching physics, mathematics or engineering without the use of such displays, if the CAS packages are available is in our opinion a disservice to the students. In a physics course, emphasis on concepts, reasoning and problem solving skills can replace drills on technical details of manipulating mathematical equations or operations required to solve routine exercises, and plots of results can underpin those concepts and critically enliven the reasoning and understanding of physics phenomena.

4. Design of the CAS-Supported Learning Environment; Examples of the Use of Symbolic Computation in Teaching and Learning Physics.

I illustrate a few aspects of teaching physics in various areas, employing Maple and/or Mathematica software packages for this purpose, via a few examples of physics teaching and learning modules. Maple was developed originally at University of Waterloo in Canada primarily to assist students in science and engineering to undertake mathematical operations on a computer in a way that a Fortran or C complier enables execution directly; although it has become a major commercial product, its devotion to an educational mission remains steadfast, and at present Maple sets a standard according to which other mathematical software can be assessed. Freely available software that is readily acquired through the Internet includes comprehensive courses, problems and applications in traditional areas of physics, such as mechanics, celestial mechanics, waves, electromagnetics, optics, quantum mechanics, mathematical and computational physics, and other applications in many areas of mathematics, science and engineering. The interrelations between science, engineering, mathematics and computing are shown in Figure 1.

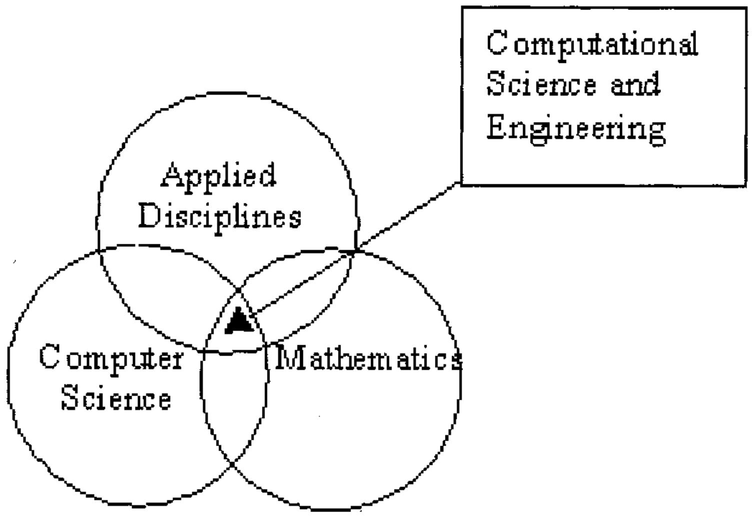


Figure 1 The definition of computational science and engineering.

Each module has three main components: lecture(s), which are part of physics or engineering courses; CAS solved related-examples; work groups and home-works. In the lecture(s), the theory is presented and examples of typical and/or real life problems are worked out using the facilities of the CAS. During the work groups, typically during the tutoring session, small groups or individual students are assigned a set of problems to solve. Students are expected to solve additional problems and to study the course text. The project total workload for a term course is about 80 hours for the average student. The main aim of the courses and the CAS-based course-supported modules is to give students a thorough understanding of fundamental concepts and approaches. Here, the groundwork is laid both for more advanced and for application-oriented technical courses. In our approach we are underway to implement or plan to develop about 15 course-supporting modules. These include: Equations of Motion, Oscillatory Motion, Electrostatics Module, Electric Circuits, Waves, Acoustics, Electromagnetic Waves, Thermodynamics, Magnetostatics, Physical Optics, Special Relativity, Quantum Phenomena, and Schrodinger Equation in One Dimension.

The first two physics teaching modules developed were form classical mechanics. One module is dedicated to the treatment of the equations of motion, while the other focused on the treatment of the oscillatory motion. Problems such as solving a system of equations and solving differential equations with constant coefficients can be readily accomplished with any CAS software and are easily handle by Maple. Among the problems studied in these modules are: pendulum and double pendulum problems, central force problem, simple harmonic motion, dumped oscillator and sinusoidally driven oscillator. In the future we intend to extend the equations of motion module to include the motion of a symmetric top and nonlinear oscillation problems (see table 1). Instructors or students can easily change the values or equations or include new graphs to include new graphical representations.

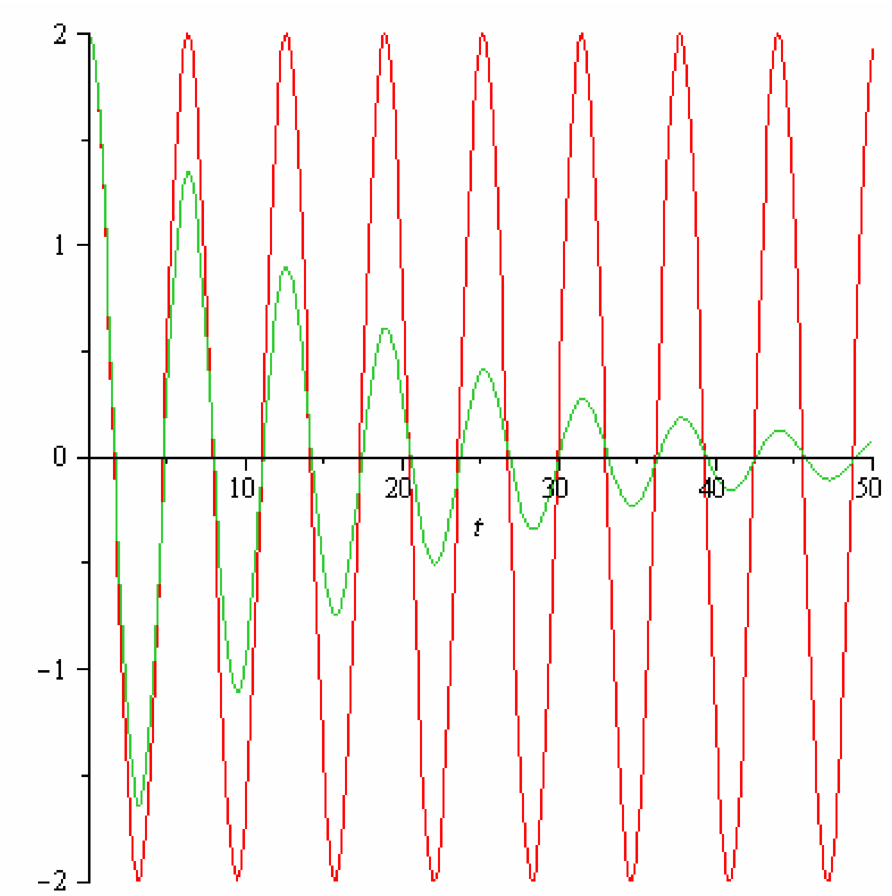
Table 1: Maple worksheet for under-damped and damped oscillator

$$Soln7 := dsolve(\{diff(x(t), t\$2) + x(t) = 0, x(0) = 2, D(x)(0) = 0\}, x(t));$$

$$Soln8 := dsolve(\{diff(x(t), t\$2) + 0.125 \cdot diff(x(t), t) + x(t) = 0, x(0) = 2, D(x)(0) = 0\}, x(t));$$

\>

\> $plot([rhs(Soln7), rhs(Soln8)], t = 0..50);$



\> $evalf(Soln7); evalf(Soln8);$

$$x(t) = 2.\ \cos(t)$$

$$x(t) = 0.1252448582\,e^{-0.06250000000\,t}\,\sin(0.9980449638\,t) + 2.\,e^{-0.06250000000\,t}\,\cos(0.9980449638\,t)$$

>

We then proceed to consider electromagnetism in static conditions. The electrostatics module is taken from the standard curriculum for first-year physics majors and from standard third year engineering electromagnetics. The module is taken as a part of a longer course on electrodynamics. Topics covered in this module include charge distributions, symmetries, Coulomb's law, Gauss' law, dipoles, multi-poles, conductors, computation of potentials with given boundaries conditions, dielectrics and polarization.

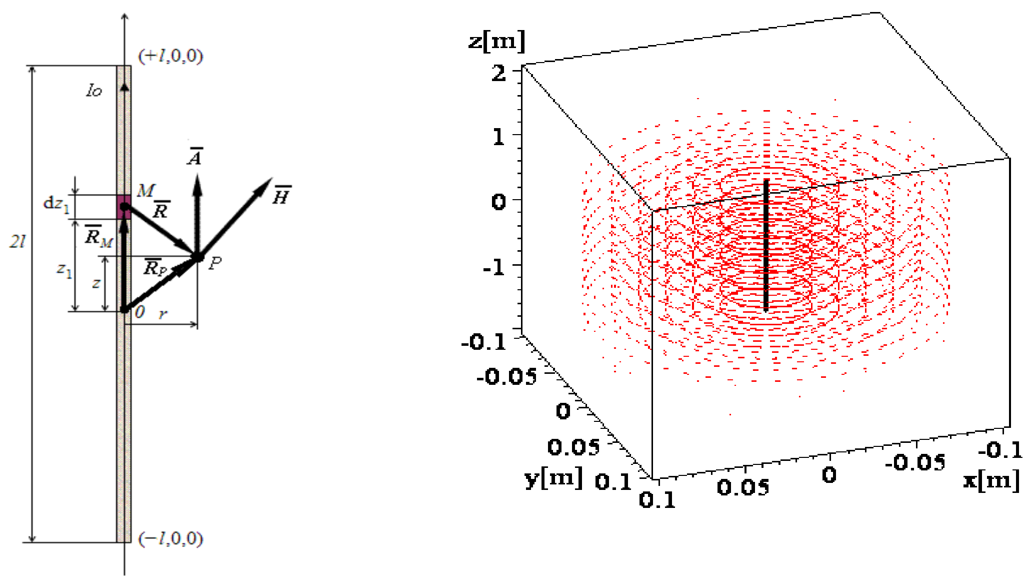


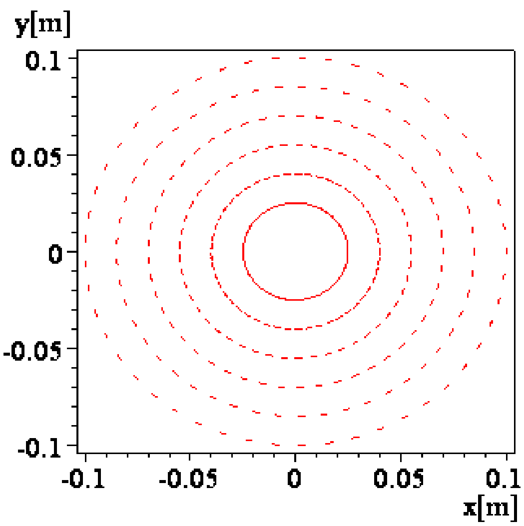
Figure 1: The straight filamentary conductor with the finite length crossed by the electric current (left panel), 3D image the magnetic field in the case of the straight filamentary conductor with the finite length (right panel).

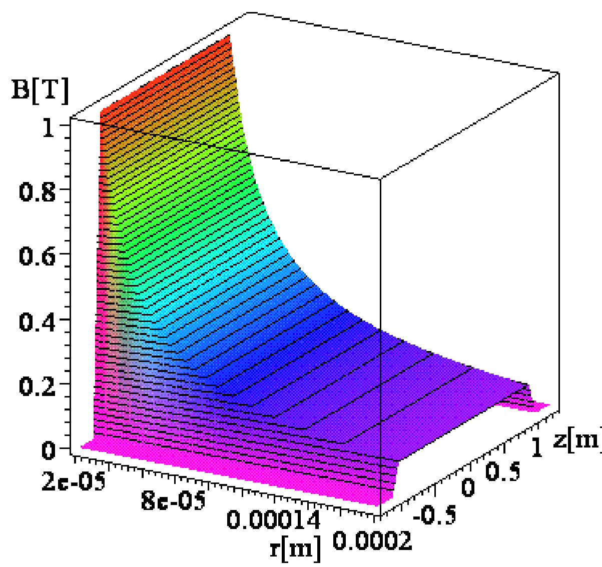
The fundamental concern of electromagnetism is to solve Maxwell's equations, and much of the course on this subject is devoted to vector calculus. To calculate an electric field and/or a magnetic field, we can perform integration directly from Coulomb's law and Biot-Savart Law, using the functions of the CAS mathematical library. For example with Maple, we can concentrate on physics, such as distinguishing the coordinates of the source point and the field point, and their separation, instead of properties of elliptic integrals. Maple provides the necessary operations such as gradient, curl/rotor and divergence in curvilinear coordinates, so one needs to spend less time on mathematics and concentrate on physics. Nowadays there is an increased tendency to use numerical methods for the electromagnetic field computation. However, the numerical approach in electromagnetic field analysis has a series of disadvantages: a) the study of the limit cases or of the result dependence of the problem parameters is

made more difficult with numerical methods, b) using numerical methods leads often to the loss of the physical meanings of the problem. These drawbacks can be eliminated by the use of symbolic methods, besides the numerical ones. The main advantages of the utilization of the symbolic computations are: a) the automatic writing of the general expressions (in any point from the space) of the magnetic field (or of the vector magnetic potential) by the adequate choice of the co-ordinates system (function of the problem symmetry) and the accurate calculation of these; b) the automatic drawing of the 2D and 3D magnetic field spectra, allowing suggestive images to be obtained;
c) the calculation of the particular solutions for which simple formulas are know, can help increase the student's confidence that the analysis was realized correctly. Some applications are now presented.

For example in Figures 2 and the magnetic field of a straight filamentary conductor of length *l*, carrying the current *Io*, in an exterior point placed at the distance *r* from the conductor. The magnitude of the magnetic field intensity is $H = [(Io)/(2\pi r)]$, in which $r = \sqrt{x^2 + y^2}$ represents the distance from the point P (in which the field is computed) to the conductor. In order to calculate the field components, the vector product of the unit vector of the current direction and the unit vector of the position vector in the xOy plane, must be computed:

$$\mathbf{H} = (\mathbf{k} \times \mathbf{r} / r)\, H. \quad (1)$$





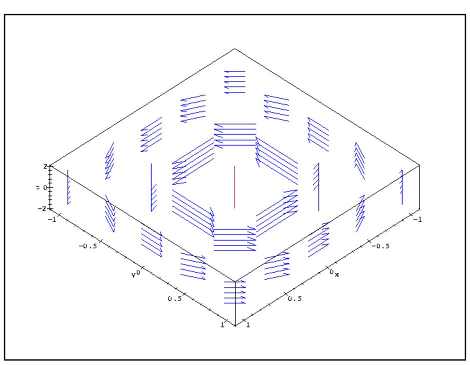


Figure 3: 2-D and 3D Magnetic flux density in an axial section in the case of the conductor with the finite length. The magnetic field visualization.

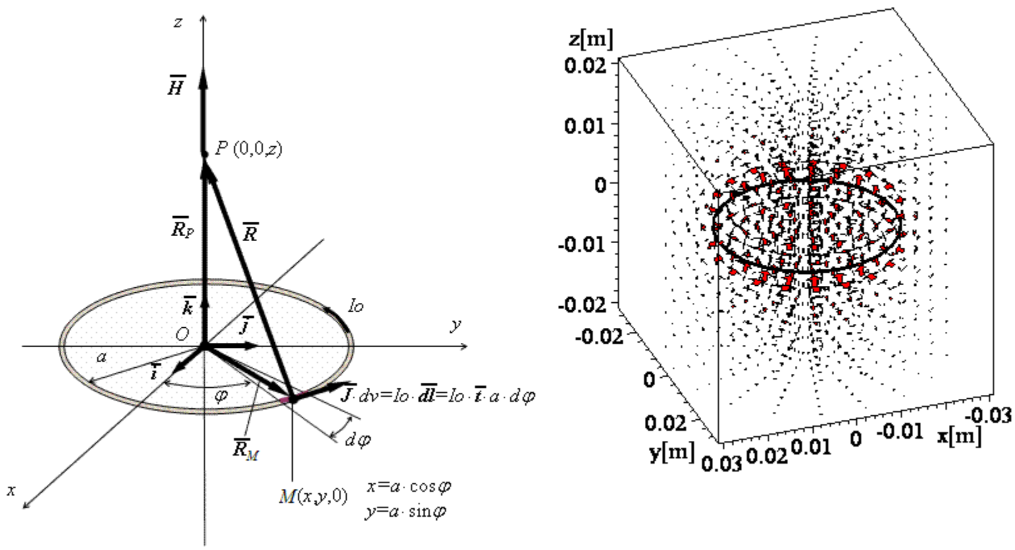


Figure 4: The straight circular single turn crossed by the electric current (left panel), 3D spectrum for the magnetic field in the case of the circular loop (right panel).

The magnetic field and the vector magnetic potential generated by the straight circular single turn crossed by the electric current is shown in Figure 4. The 3D magnetic field spectrum (fig. 5) and the 3D variations of the magnetic flux density in a parallel plane with the turn placed to a distance z and in an axial section (fig. 7, 8) were plotted on the basis of the obtained solutions. The values of the parameters are: electric current intensity $I = 100$ A, turn radius $a = 2$ cm. These examples show the advantages of CAS software packages in visualization of electromagnetic fields, which significantly enhance the student understanding of such phenomena.

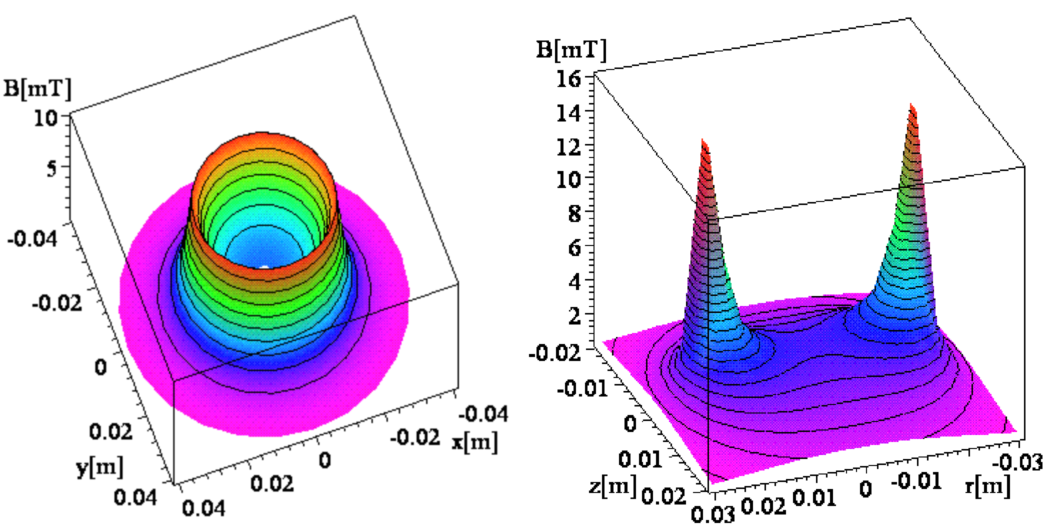


Figure 5: 3D variation of the magnetic flux density in a parallel plane with the turn placed to a distance *z* in the case of the straight circular single turn (left panel), 3D variation of the magnetic flux density in an axial section in the case of the straight circular single turn.

The module referring to the electric circuit focuses on two main topics: a) DC circuit, including the RC, RL and RLC circuits; and b) on the AC circuits. The solving electric circuits involve the applications of solving a system of algebraic and differential equations, a topic similar to oscillatory motion, which is one of the strong capabilities of every CAS software, and in particular of Maple. In this module we also use Maple's capability of complex numbers to treat problems of alternating-current circuits. Figure 7 and Table 2 are showing the Maple solving of RLC circuits.

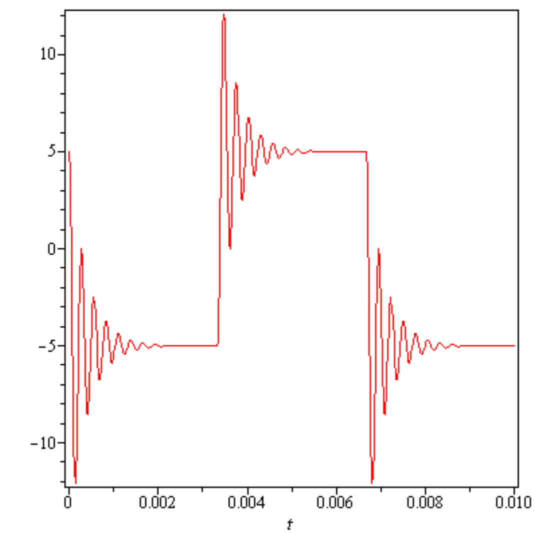


Figure 7: RLC Circuit waveforms

In the modules of waves and of optics, which are under way to be developed, because we deal with function containing both spatial and temporal components, we will take advantage of Maple to produce animations that allow visualization. The content of these modules span from simple motion and standing waves to advanced optics, such as a dispersion relations, which is important in quantum waves, animations illuminating both the spatial and temporal properties of waves. Physical optics involves the addition of waves: we approach this topic using Maple's graphic ability to display the final amplitude of waves in various combinations. Electromagnetic waves module, also in process to be developed includes the first stage study of the dipole radiation and the synchrotron radiation problem. Other topics will be added soon.

5. Conclusions and Future Work

The paper has reported on the development of a set of teaching and learning modules using symbolic computation for university physics courses. The goal was to support students in gaining intuitive understanding of physical situations, solution methods, or the relations between them and to help them to get inside of less intuitive phenomena, such as electromagnetic field phenomena or quantum theory.

These experiments are expected to improve students understanding of physical situations and to strengthen the relations they see between the solution methods and the situation features.((Cam repeti acelasi lucru cu astead doua propozitii.)) Among the distinctive features of the CAS modules are the use of precise language for specifying problems, visualization support and symbolic and numerical support for solving problems. The future work will consists in the improvement and extension of the already developed modules, the design and implementation of new modules. Long term goal is the development and design an e-learning version of the CAS modules.

References

1. Cohen H.I. and J.P. Fitch, Uses Made of Computer Algebra in Physics, *J. Symbolic Computation*, Vol. 11, pp. 291-305, 1991.

2. Corbett A.T. and J. R. Anderson, LISP intelligent tutoring system: *Research in skill acquisition, In Computer-Assisted Instruction and Intelligent Tutoring System* (eds J.H. larkin & R.W. Chabay), pp. 73-109, 1992.

3. Glasser R. and M. Bassok, Learning theory and study of instruction, *Annual Review of Psychology*, Vol.40, pp. 631-666, 1989.

4. Gradshteyn I.S. and I.M. Ryzhik, *Tables of Integrals, Series and Products*, Academic Press, 1965.

5. Coddington, *Theory of differential equations,* Wiley, 1997.

6. MacDonald N., Introducing physics students to symbolic computation, *Eur. J. Phys*., Vol. 12, pp.1015, 1991

7. Rutherford F.J. and A. Ahlgren, *Science for All Americans*, Oxford University Press, New York, 1991.

8. Wang F.Y., *Physics with Maple*, Wiley, 2005

9. Mathematica at home: http://www.wolfram.com

10. Maple at home: http://www.maplesoft.com

11. AXIOM at NAG: http://www.nag.co.uk/symbolic/AX.html

12. MuPAD at home: http://www.mupad.de

13. REDUCE in Cologne: http://www.rrz.uni-koeln.de/REDUCE/

14. Matlab at home: http://www.mathworks.com

15. Mathcad at home: http://www.mathsoft.com

16. Visual Numerics at home: http://www.vni.com

17. Scientific Workplace: http://www.tcisoft.com

18. MathView at home: http://www.cybermath.com

19. IBM's Techexplorer: http://www.alphaworks.ibm.com/formula/techexplorer/

20. MacTutor's Pi-page: http://wwwgroups.dcs.stand.ac.uk/~history/HistTopics/Pi_through_the_ages.html

Table 2: Worksheet for the double pendulum

```
> x1 := l1*sin(theta1(t));

> y1 := l1*cos(theta1(t));

> x2 := x1+l2*sin(theta2(t));

> y2 := y1+l2*cos(theta2(t));

> T := (1/2)*m1*((diff(x1, t))^2+(diff(y1, t))^2)+(1/2)*m2*((diff(x2, t))^2+(diff(y2, t))^2);

> T := combine(T);

> V := -m1*g*y1-m2*g*y2;

> L := T-V;

                 2                  2

 1    2 / d       \  1    2 / d       \
```

```
 - m1 l1  |--- theta1(t)|  + - m2 l1  |--- theta1(t)|
 2        \ dt          /    2        \ dt          /

          / d           \    / d           \
   + m2 l1 |--- theta1(t)| l2 |--- theta2(t)| cos(theta1(t) - theta2(t))
          \ dt          /    \ dt          /

                        2
      1      2 / d           \
   + - m2 l2  |--- theta2(t)|  + m1  g l1 cos(theta1(t))
      2        \ dt          /

   + m2 g (l1 cos(theta1(t)) + l2 cos(theta2(t)))
```

```
> L1 := subs({diff(theta1(t), t) = var2, diff(theta2(t), t) = var4, theta1(t) = var1, theta2(t) = var3}, L);
> Epr11 := diff(L1, var2);
> Epr12 := diff(L1, var1);
> Epr13 := subs({var1 = theta1(t), var2 = diff(theta1(t), t), var3 = theta2(t), var4 = diff(theta2(t), t)}, Epr11);
> Epr14 := subs({var1 = theta1(t), var2 = diff(theta1(t), t), var3 = theta2(t), var4 = diff(theta2(t), t)}, Epr12); Epr15 := diff(Epr13, t);
                              0
> Eq16 := Epr15-Epr14 = 0;
> Eq17 := collect(Eq16, diff);
> Epr21 := diff(L1, var4);
> Epr22 := diff(L1, var3);
> Epr23 := subs({var1 = theta1(t), var2 = diff(theta1(t), t), var3 = theta2(t), var4 = diff(theta2(t), t)}, Epr21);
```

```
> Epr24 := subs({var1 = theta1(t), var2 = diff(theta1(t), t), var3 = theta2(t), var4 = diff(theta2(t), t)}, Epr22);

> Epr25 := diff(Epr23, t);

> Eq26 := Epr25-Epr24 = 0;

> Eq27 := collect(Eq26, diff);

> m1 := 0.5e-1; m2 := 0.5e-1; l1 := .5; l2 := .5; g := 9.8;
```

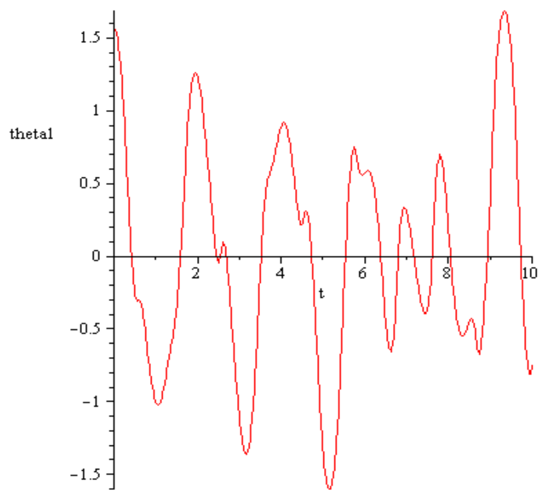
0.05

0.05

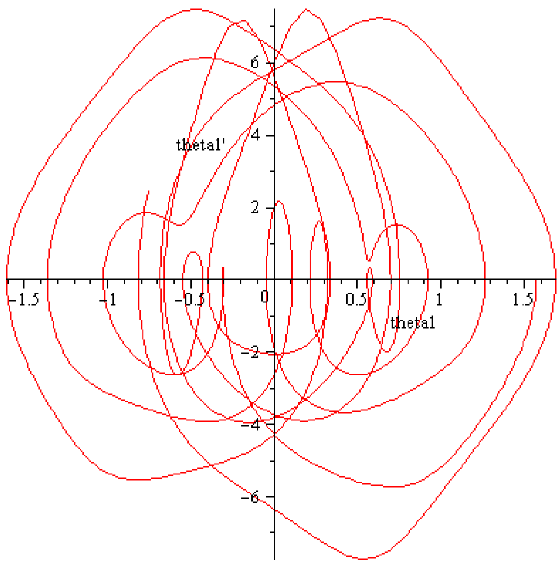
0.5

0.5

```
> ini := theta1(0) = Pi/(2.0), (D(theta1))(0) = 0, theta2(0) = Pi/(4.0), (D(theta2))(0) = 0;

> Eq75 := dsolve({ini, Eq17, Eq27}, {theta1(t), theta2(t)}, numeric, output = listprocedure);

> with(plots); with(plottools);

> odeplot(Eq75, [t, theta1(t)], 0 .. 10, numpoints = 200);
```




```
> odeplot(Eq75, [theta1(t), diff(theta1(t), t)], 0 .. 10, numpoints = 800);
```



> noffm := 100; divs := 10;

100

10

> for i from 0 to noffm do x1 := l1*sin(rhs(Eq75[2](i/divs))); y1 := -l1*cos(rhs(Eq75[2](i/divs))); x2 := x1+l2*sin(rhs(Eq75[4](i/divs))); y2 := y1-l2*cos(rhs(Eq75[4](i/divs))); rod[i] := curve([[0, 0], [x1, y1], [x2, y2]]); ms1[i] := disk([x1, y1], 0.2e-1, color = red); ms2[i] := disk([x2, y2], 0.2e-1, color = blue); anima[i] := display({ms1[i], ms2[i], rod[i]}) end do;

> for i from 0 to 5 do display(anima[i], insequence = true, scaling = constrained, axes = none) end do;

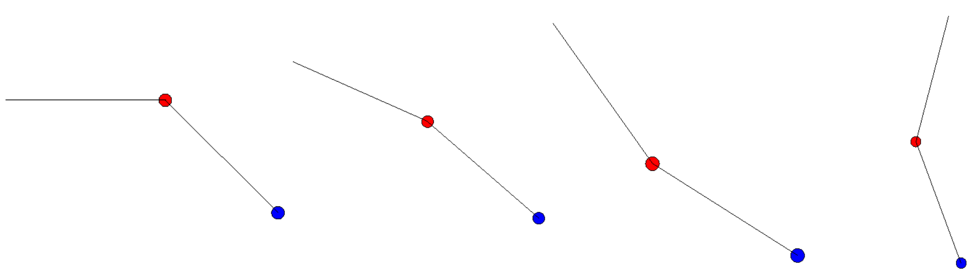


Table3: Maple worksheet for solving RLC Circuits

> $V0 := 5;\ P := \frac{1}{300};\ C := 0.19\mathrm{e}{-6};\ R := 50;\ L := 10\mathrm{e}{-3};$

> $Eq1 := \frac{Q1(t)}{C} + diff(Q1(t), t) \cdot R + L \cdot diff(Q1(t), t\$2) = -V0;$

> $Soln1 := dsolve(\{Eq1, Q1(0) = C \cdot V0, \mathrm{D}(Q1)(0) = 0\}, Q1(t));$

> $assign(Soln1);$

> $Eq2 := \frac{Q2(t)}{C} + diff(Q2(t), t) \cdot R + L \cdot diff(Q2(t), t\$2) = V0;$

> $Soln2 := dsolve(\{Eq2, Q2(P) = -C \cdot V0, \mathrm{D}(Q2)(P) = 0\}, Q2(t));$

> $assign(Soln2);$

> $Eq3 := \frac{Q3(t)}{C} + diff(Q3(t), t) \cdot R + L \cdot diff(Q3(t), t\$2) = -V0;$

> $Soln3 := dsolve(\{Eq3, Q3(2 \cdot P) = C \cdot V0, \mathrm{D}(Q3)(2 \cdot P) = 0\}, Q3(t));$

> $assign(Soln3);$

> $with(plots);$

> $p1 := plot\left(\frac{Q1(t)}{C}, t = 0..P\right);$

> $p2 := plot\left(\frac{Q2(t)}{C}, t = P..2 \cdot P\right);$

> $p3 := plot\left(\frac{Q3(t)}{C}, t = 2 \cdot P..3 \cdot P\right);$

> $display([p1, p2, p3], axes = BOXED);$