U.P.B. Sci. Bull., Series A, Vol. 68, No. 1, 2006

OPENPH - NUMERICAL PHYSICS LIBRARY

G. MILESCU, G. NOAJE, Fl. POP*

Fizica numerică a căpătat o importanţă deosebită în ultimele decenii, eficienţa sa fiind motivată şi sustinută de creşterea puterii de calcul a computerelor. Această lucrare prezintă un concept care va fi dezvoltat în urmatorii ani: OpenPh. OpenPh este o biblioteca de fizică numerică care foloseşte atât avantajele programelor „open source" cât şi avantajele lucrului cu MATLAB. Scopul ei este de a aduce instrumente pentru găsirea soluţiilor numerice şi grafice pentru diferite probleme ale fizicii. OpenPh are o structură modulară permiţând utilizatorului să adauge noi module celor existente şi să îşi creeze propriile module în conformitate cu nevoile sale, putând fi în acest fel extinsă oricât. Modulele OpenPh sunt implementate folosind motorul MATLAB deoarece acesta este cea mai buna soluţie folosită în inginerie şi ştiinţă, oferind o gama largă de metode optimizate pentru a îndeplini chiar si cele mai dificile sarcini.

Numerical physics has gained a lot of importance in the last decade, its efficiency being motivated and sustained by the growth of computational power. This paper presents a concept that is to be developed in the next few years: OpenPh. OpenPh is a numerical physics library that makes use of the advantages of both open source software and MATLAB programming. Its aim is to deliver the instruments for providing numerical and graphical solutions for various physics problems. It has a modular structure, allowing the user to add new modules to the existing ones and to create its own modules according to its needs, being virtually unlimited extendable. The modules of OpenPh are implemented using MATLAB engine because it is the best solution used in engineering and science, providing a wide range of optimized methods to accomplish even the toughest jobs.

Keywords: numerical physics, MATLAB, quantum physics, classical mechanics, module, photoelectric effect, Schrödinger equation, vibration of a string

Introduction

Numerical physics has gained a lot of importance in the last decade, its efficiency being motivated and sustained by the growth of computational power. The use of numerical physics libraries eases the understanding of various phenomena and the solving different types of problems. This is the reason why more and more numerical physics libraries come to life, each of them bringing

---

* Student, student, Engineer, Automatic Control and Computers Faculty, University "Politehnica" of Bucharest, ROMANIA

different particularities. The advantage of OpenPh among other libraries it the fact that it makes use of the advantages open source software, allowing each user to bring its contribution to the development of the library. This concept, combined with its modular structure and the use of Matlab engine, differentiate OpenPh from other libraries.

Matlab environment is relatively new and therefore physics libraries are not too common. Lots of physics libraries are developed in old programming languages like Fortran or even Basic. On the Internet there are some libraries, but almost all of them treat optics ("Plasma Physics Library" www.pppl.gov/library/july00.html, Abos Physics library bibsrv.physik.rwth-aachen.de/abos.html). OpenPh library offers very useful routines for other fields of physics.

The foundation for this physics library is the well-know computational engine Matlab. Why Matlab? Because it is the award-winning solution used in engineering and science providing a wide range of means to accomplish even the toughest jobs. It modularity and facile expansion allows every user to develop its own modules and application to fulfill its needs. We have chosen Matlab because, in many applications, we need to find the derivate or integral of a function. There is not always an analytical form for the integral and in fact the function itself may not be exactly known. In these cases the derivative or integral can only be determined numerically. Although many sophisticated techniques exist, we can also easily use the array manipulation tools in Matlab to make the calculations with a reasonable precision.

The programming techniques used in Matlab reflect its scripting language power. The symbolic tools, plotting options and capability to reduce execution time by vector compute are used to approach the physics problems.

OpenPh library was started like a modular library. The user can use the functions provided by OpenPh to write other functions. The simulation of a simple experiment (like photoelectrical effect or Carbon disintegration) is easy to build with our functions.

The library presented is far from being complete, but due to its modularity and scalability it allows virtually unlimited extension. Currently it includes functions for solving problems from two different domains of the physics: mechanics and quantum physics. In addition, it includes some general functions basically oriented on mathematics. These functions can be used by any of the specialized modules in order to simplify calculations or as stand-alone modules.

The two modules currently available in the library deliver the following specific functions:

- *Classical Mechanics:* uniform circular motion, forced damped harmonic oscillations, simple harmonic oscillator (pendulum), vibration of a fixed-fixed string

- *Quantum physics:* photoelectric effect, simulated radioactive decay of carbon-11, Schrödinger equation - particle in a box

Classical mechanics

Classical mechanics is one of the two major sub-fields of study in the mechanics, which is concerned with the motions of bodies, and the forces that cause them. The library offers tools to analyze and calculate some common problems in classical mechanics.

1. Uniform circular motion

Circular motion is movement with constant speed around a circle (figure 1). A function for circular motion was included in the library in order to represent an intuitive path. (Fig. 2)

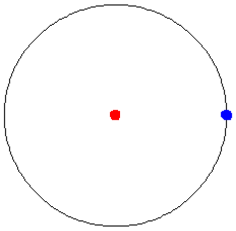


Fig. 1 – Circular motion path

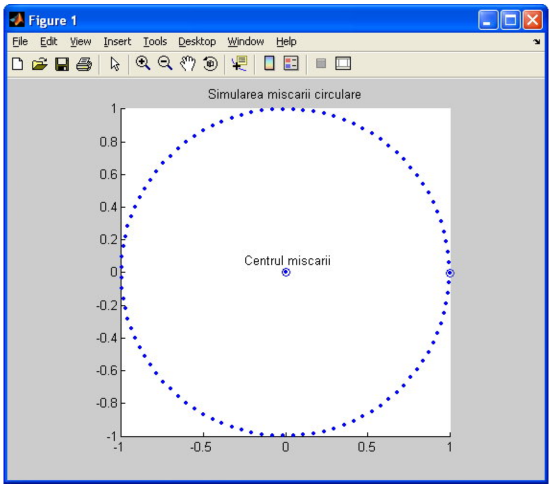


Fig. 2 – Uniform circular motion simulation

2. Forced damped harmonic oscillations

Derived from the general equation of the forced damped harmonic oscillator, simpler oscillators exist. The library offers a set of powerful functions allowing the simulation and calculating different elements of an oscillator such as amplitude, angular frequency and acceleration. Each of these elements can be graphically represented. Fig. 3 shows such an example.

Furthermore, there are used two different ways of computing the elements of the oscillator, that's why a special function has been implemented to make four plots comparing the small amplitude solution with the numerical solution. (Fig. 5). If we analyze the results, we can observe that for small amplitudes (such as 0.5 or 1 units) the two solutions dovetail. If the amplitude increases (2 or 4 units) then the small amplitude solution performs an error that can no longer be neglected, while the numerical keeps being exact.

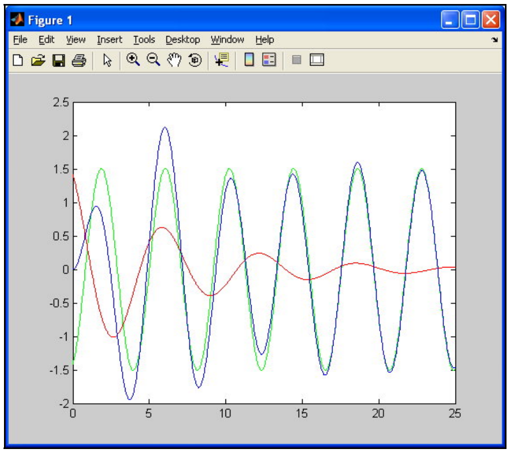


Fig. 3 – Amplitude, angular frequency, Acceleration of an oscillator

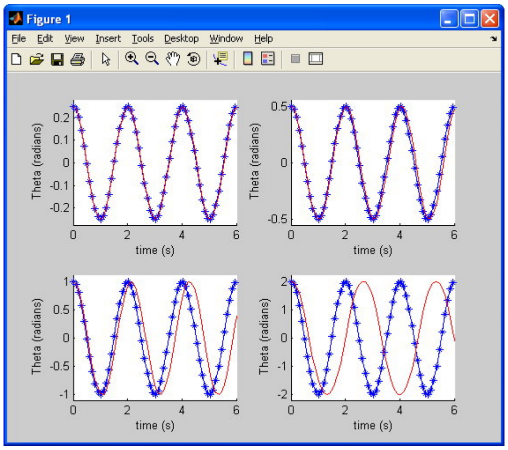


Fig. 4 – Comparison graphs

3. Vibration of a fixed-fixed string

When the end of a string is fixed, the displacement of the string at that end must be zero. A transverse wave traveling along the string towards a fixed end will be reflected in the opposite direction. When a string is fixed at both ends, two waves traveling in opposite directions simply bounce back and forth between the ends.

A string which is fixed at both ends will exhibit strong vibrational response only at the resonance frequencies. At any other frequencies, the string will not vibrate with any significant amplitude. The resonance frequencies of the fixed-fixed string are harmonics (integer multiples) of the fundamental frequency (n=1).

Fig. 5 - Standing wave in stationary medium
(the red dots represent the wave nodes)

This equation represents a standing wave. There will be locations on the string which undergo maximum displacement (antinodes) and locations which to not move at all (nodes). In fact, the string may be touched at a node without altering the string vibration. (Fig. 5)

The library has a function that shows the motion of such string in a very realistic manner. (Fig. 6)

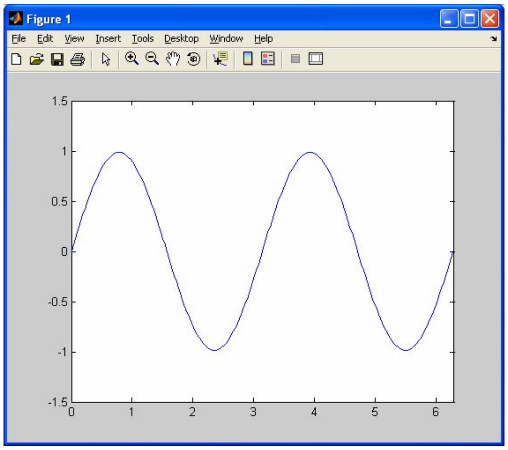


Fig. 6 – String vibration simulation

Beside the specialized modules there are also some generic functions such as:

- function that generates Fahrenheit-Celsius conversion table
- function that generates comparison table between factorial and Stirling approximation

Quantum physics

Quantum mechanics is a fundamental physical theory which extends and corrects Newtonian mechanics, especially at the atomic and subatomic levels. The library offers tools to analyze and calculate some common problems in quantum theory.

1. Photoelectric effect

In order to present the photoelectric effect the circuit in figure 7 is realized. The library function calculates the voltage when no current is passing through the multimeter.

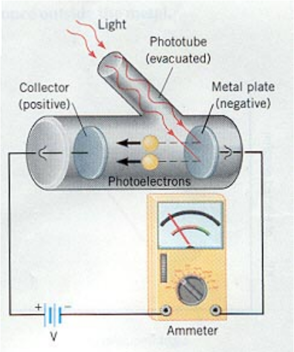


Fig. 7 – Experiment for the photoelectric effect

2. Simulated radioactive decay of carbon-11

Radioactive decay is the set of various processes by which unstable atomic nuclei (nuclides) emit subatomic particles.

The decay of an unstable nucleus (radionuclide) is entirely random and it is impossible to predict when a particular atom will decay. However, it is equally likely to decay at any time. Therefore, given a sample of a particular radioisotope, the number of decay events expected to occur in a small interval of time *dt* is proportional to the number of atoms present.

Particular radionuclides decay at different rates, each having its own decay constant (λ). The solution to this equation satisfies the following decay law: $N(t) = N_0 e^{-\lambda t}$. [1]

This function is only an approximate solution, for two reasons. Firstly, the exponential function is continuous, but the physical quantity N can only take positive integer values. Secondly, because it describes a random process, it is only statistically true. However, in most common cases, N is a very large number and the function is a good approximation. [1]

The library function tries to express this approximation presenting the theoretical function and the randomly generated number of decayed nuclei for Carbon 11 atom. (Fig. 8, 9)

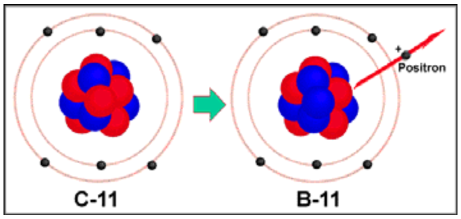


Fig. 8– Radioactive decay of carbon-11 (scheme)

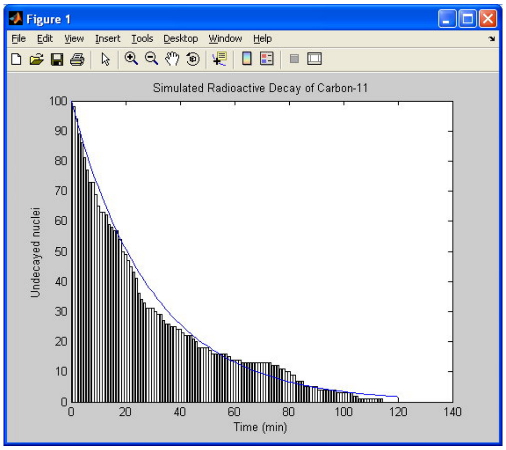


Fig. 9 - Simulated radioactive decay of carbon-11

3. Schrödinger equation – free particle in a box

In physics, the particle in a box (or the square well) is a simple idealized system that can be completely solved within quantum mechanics. It is the situation of a particle confined within a finite region of space (the box) by an infinite potential that exists at the walls of the box. The particle experiences no forces while inside the box, but is constrained by the walls to remain in the box.

The quantum behavior in the box includes: energy quantization (it is not possible for the particle to have any arbitrary definite energy; only discrete definite energy levels are allowed), zero-point energy (the lowest possible energy level of the particle, called the zero-point energy, is nonzero), states (the Schrödinger equation predicts that for some energy levels there are positions at which the particle can never be found).

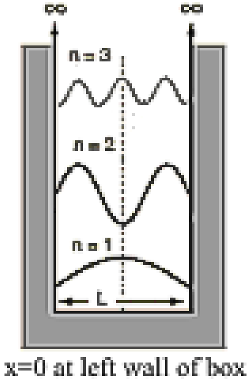
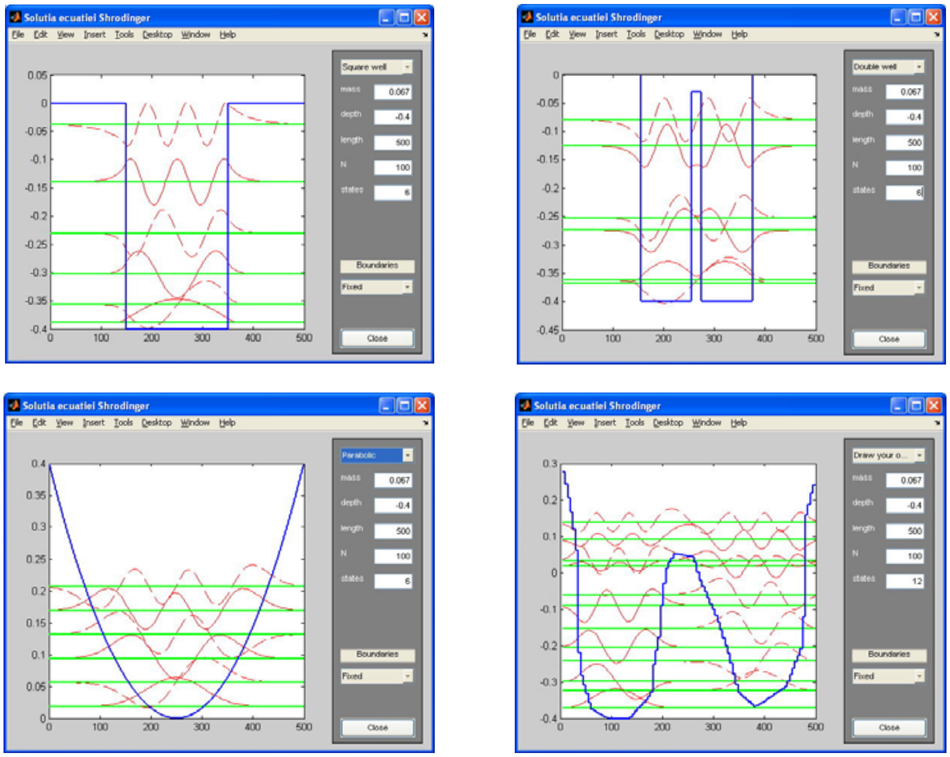


Fig. 10 – Particle in a box with infinite walls

Analytical solutions of the time-independent Schrödinger equation can be obtained for a variety of relatively simple conditions. One of these simple conditions is the particle in a 1-dimensional box. (figure 10).

The library function gives the solution to the equation for four different types of boxes (square well, double well, parabolic, free-hand: the box can be defined by the user). (Fig. 11)



Square well
mass 0.067
depth -0.4
length 500
N 100
states 6
Fixed

Double well
mass 0.067
depth -0.4
length 500
N 100
states 6
Fixed

Parabolic
mass 0.067
depth -0.4
length 500
N 100
states 6
Fixed

Draw your o...
mass 0.067
depth -0.4
length 500
N 100
states 12
Fixed

Fig. 11 - Particle in a 1-dimensional box

Conclusions

The idea of a numerical physics library is not new. It has been implemented in various libraries in the past few years. What OpenPh brings extra are its modularity and the benefits of open source software. The modules of OpenPh are created in such a manner that allows them to be “connected”, creating a new module by using a combination of existing modules.

If one of the users creates an application that can be integrated in OpenPh, this application could be used to improve or create new modules.

The current state of the library offers a wide range of tools for solving different types of problems. It has a user-friendly interface and it is extremely easy to use.

Relying on the powerful Matlab engine, functions can work with large sets of data, having practical applications.

Graphical representations of the theoretical models offer a better perspective over the simulated phenomena. Graphics are much more intuitive elements for analysis than the formulas or numbers.

OpenPh is to be continued in the years to come. The community that is already created is maybe the biggest advantage of this library: that it is continuously evolving, new modules being created and the existing ones being improved. Anyone can be a member of the community, making use of software written by others and sharing its own results.

REFERENCES

1. *Richard P. Feynman, Robert B. Leighton, Matthew Sands* “Lectures on Physics” (complete and definitive issue), Pearson, Addison-Wesley, 2nd edition, 2005
2. *David Halliday, Robert Resnick, Jearl Walker* “Fundamentals of Physics”, Wiley, 2004
3. *Dan Iordache* “Numerical Physics” (in Romanian), vol.1, Man-Dely Printing House, Bucharest, 2004
4. *I. M. Popescu* “Physics”, vol.1 (in Romanian), Didactic and pedagogical printing house, Bucureşti, 1981
5. *Won Young Yang, Wenwu Cao, Tae-Sang Chung, John Morris* “Applied Numerical Methods Using MATLAB”, John Wiley & Sons, New Jersey, 2005
6. *Patrick Marchand, O. Thomas Holland*, “Graphics and GUIs with MATLAB (Third Edition)”, Chapman & Hall/CRC, 2002
7. *H. B. Wilson, Louis H. Turcotte, David Halpern*, “Advanced Mathematics and Mechanics Applications Using MATLAB (Third Edition)”, Chapman & Hall/CRC, 2003
8. *MATLAB web page*: http://www.mathworks.com