An open source virtual laboratory for the Schrödinger equation

To cite this article: Edgar Figueiras et al 2018 Eur. J. Phys. 39 055802

View the article online for updates and enhancements.

Related content
- Designing an ultrafast laser virtual laboratory using MATLAB GUIDE
  F Cambronero-López, A I Gómez-Varela and C Bao-Varela
- Waves in hyperbolic and double negative metamaterials including rogues and solitons
  A D Boardman, A Alberucci, G Assanto et al.
- Towards a quantum time mirror for non-relativistic wave packets
  Philipp Reck, Cosimo Gorini, Arseni Goussev et al.

IOP ebooks
Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.
Start exploring the collection - download the first chapter of every title for free.
An open source virtual laboratory for the Schrödinger equation

Edgar Figueiras¹, David Olivieri¹ ², Angel Paredes² ³  and Humberto Michinel²

¹ Departamento de Linguaxes e Sistemas Informáticos, Universidade de Vigo. As Lagoas s/n, Ourense, E-32004, Spain
² Applied Physics Department, School of Aeronautic and Space Engineering, Universidade de Vigo. As Lagoas s/n, Ourense, E-32004, Spain

E-mail: efgomez@esei.uvigo.es, olivieri@uvigo.es, angel.paredes@uvigo.es and hmichinel@uvigo.es

Received 30 April 2018, revised 24 May 2018
Accepted for publication 1 June 2018
Published 27 June 2018

Abstract
A simple Python-based open source software library for the numerical simulation of the linear or nonlinear time-dependent Schrödinger equation in one and two dimensions is presented. The integration is performed using a first-order split-step pseudospectral method, relying on the fast Fourier transform. The software library could be useful for undergraduate courses in elementary quantum mechanics, wave optics and computational physics. It could also be of interest for graduate students working with nonlinear waves, in frameworks such as laser beam propagation in nonlinear optical materials, matter waves within ultracold gases, dark matter or superfluid dynamics, among others. The discussion is complemented by solved examples and suggestions for educational applications of the code.

Keywords: virtual laboratory, Schrödinger equation, beam propagation method, open source, Python

(Some figures may appear in colour only in the online journal)
1. Introduction

The emergence of several high-level software languages has greatly simplified the art of writing computer programmes for numerical simulation. As such, interpreted languages such as Python provide rapid development through streamlined syntax, yet compiled-time speed through third party modules (i.e. with Numpy, Scipy, and Numba) and C-extensions. Because of this paradigm shift away from writing numerical code in strongly typed compiled languages, such as C/C++, efforts in teaching and learning computational physics can focus on the physics and numerical algorithms and not syntactical details of software development.

Combined with visualisation tools, interactive software environments, or virtual laboratories, can be constructed for designing and performing simulated experiments. Apart from the merits of learning the intricacies of computational algorithms, a virtual laboratory is a useful complement to a physical laboratory for student training [1], since it is more affordable, and in some cases, requires less time to illustrate key physics ideas [2], and can be used in distance learning [3]. Virtual laboratories can also play an important role in the understanding of abstract concepts and the ramification of complex equations, helping students gain intuition that may complement the learning from real experiments [1, 2].

An area of physics where such virtual laboratories have a particular appeal is quantum mechanics and contemporary topics in nonlinear quantum dynamics. In general, few real systems exist where the Schrödinger equation can be solved analytically, so that numerical computation is necessary. Despite the large amount of textbooks and literature devoted to numerical computation, the details of numerical methods for the Schrödinger equation, especially with nonlinear potentials, is a nontrivial task and often not accessible to the novice student. In this context, we present a free and open source software library [4], designed for educational purposes, that illustrates key methods for solving quantum problems described recently in the scientific literature. In particular, with this library students can numerically solve the time-dependent Schrödinger equation in one or two dimensions and visualise the results. Moreover, since the software is presented as a Python library, the student can easily extend the examples provided to adapt to innumerable configurations.

In the Schrödinger equation, the function \( \psi(x, t) \) describes the evolution of a quantum wave under the action of a potential, \( V \). If \( V \) is \( \psi \)-independent, the equation is linear, representing typical cases discussed in elementary quantum mechanics courses. The same linear equation describes paraxial wave dynamics used in beam optics, that is applicable to laser light propagation [5]. When \( V \) depends on \( \psi \), the resulting Schrödinger equation is nonlinear and is a paradigmatic model for describing a wide variety of nonlinear waves (NLWs) phenomena. The importance and prevalence of NLWs in nature cannot be understated, since they appear in a wide amount of systems, including nonlinear optics [6], biology [7], superfluidity [8], cosmology [9, 10], amongst others.

Because of their deep implications of NLWs modern physics, exploring their often counterintuitive dynamics is a requisite experience for contemporary undergraduate and graduate students. For this, we developed an easy-to-use numerical software virtual laboratory tool and library, based upon a general purpose language, that is ideal for introducing basic and advanced concepts of NLWs propagation. Our library is specifically designed as an educational tool to help provide an important first step towards understanding the numerical algorithms and behaviour or NLW found in current research literature.

Many modern textbooks and articles describe in great detail the design and implementation of numerical algorithms to solve the time-dependent Schrödinger equation, e.g. [11, 12]. Also, several ready to use virtual laboratories, written for different software languages and environment, treat this equation in particular situations, e.g. [13, 14]. Here, we describe our virtual laboratory library and how it deals with many specific practicalities of
simulating NLWs, that are often only briefly discussed in the research literature. In particular, the design meets the following goals:

- Wide accessibility: it runs on any of the major operating systems and only relies on free software.
- Easy to install and use: the library and examples can be readily installed and executed (a step-by-step tutorial is provided [4]) with relatively modest effort.
- Flexibility: the library is modular and easy to use. Some students may wish to simply explore the physical results of examples by tuning the set of input parameters in initial conditions. However, the library and examples can be readily extended by advanced undergraduate or graduate students to model new systems.
- Simplicity: complex programming structures and methods (i.e. object oriented design patterns and data structures) that may be obscure for a beginner have been purposely excluded. Also, each line of code in the library and example scripts is annotated so that the logic can be followed at multiple levels.
- Direct visualisation of results: during and after the computation, the software library provides functions to produce images and animations to visualise the evolution of the wave function (notice that this may help in addressing typical problems that students find in this respect [15]).

2. On the split-step algorithm and its implementation in Python

The Schrödinger equation, written in terms of dimensionless quantities, is

\[ i\frac{\partial \psi}{\partial t} = -\frac{1}{2} \nabla^2 \psi + V\psi, \]  

A numerical solution of equation (1) involves specifying the initial condition \( \psi(x, t = 0) \) and the potential, \( V(x, t, \psi) \). Our software library solves equation (1) for problems in both one and two dimensions (1D and 2D), where in 1D, \( x = x \), \( \nabla^2 = \partial_x^2 \), while in 2D, \( x = (x, y) \), \( \nabla^2 = \partial_x^2 + \partial_y^2 \).

2.1. The beam propagation method (BPM)

An efficient and often used numerical approximation technique for solving equation (1), particularly for the nonlinear case, is the split-step Fourier method, also referred to as the BPM in the context of optics. Originally developed for studying the propagation of intense laser beams through the atmosphere [16], the method (and further higher order approximations) has since been described many times in the literature (e.g. [17]). The essential observation behind the BPM algorithm is that the right-hand side of equation (1) can be split in two terms: one containing derivatives and the other derivative-free. In terms of operator splitting, \( i\partial_t \psi = (\hat{L} + \hat{V}) \psi \). A solution to this equation is \( \psi(x, t) = e^{-i(t(L+V))} \psi(x, 0) \). In this way, the spatio-temporal evolution associated with the Laplacian term is computed in Fourier space, while the potential term is computed in position space. It follows that \( \psi(x, t + dt) = e^{-i dt} e^{-i\hat{V} dt} \psi(x, t) + O(dt^2) \), where the higher order terms \( O(dt^3) \) are related to operator commutation.

The basic form of the BPM algorithm is expressed as

\[ \psi(x, t + dt) \approx \mathcal{F}^{-1} [e^{i dt (2\pi i k)^2} \mathcal{F}[e^{-i\hat{V} dt} \psi(x, t)]], \]

where \( \mathcal{F} \) and \( \mathcal{F}^{-1} \) are the Fourier transform and its inverse, respectively. Implementing the above equation as a numerical algorithm involves the following procedure:
I. Multiply the input function by $\psi \leftarrow e^{-idV\psi}$. 

II. Perform the Fourier transform.

III. Multiply the result by $e^{\frac{id\omega}{2\pi k}t}$. 

IV. Calculate the inverse Fourier transform.

V. Advance the time step by $t \leftarrow t + dt$

Successive application of the previous steps advances $\psi(x, t = 0)$, at time $t = 0$, to $\psi(x, t)$ at some time $t$. The algorithm converges provided that $dt$ is small: $(\hat{L} + \hat{V})dt \ll 2\pi$ for each step. The conservation of $\int |\psi|^2 dx$ is automatically guaranteed by the algorithm, provided that $V$ is real. A practical method to verify the precision of the algorithm for a particular time difference and grid size is to compare the values of $\psi$ obtained by varying the values of $dt$ and the spatial grid spacing.

A complete specification of equation (1) requires that the initial and boundary conditions $\psi(x \in \mathcal{B}, t)$ be fixed, where $\mathcal{B}$ is the spatial boundary of the computational domain. Because the Fourier transform is defined implicitly on a periodic domain, the boundaries with the split-step method must be treated explicitly; a wave packet reaching a periodic boundary of the computational domain would re-enter on the opposite side. In some cases, such periodic conditions are not convenient. Imagine, for example, an optical system in which part of the energy is radiated away from the core of the optical material, resulting in net energy loss. By using periodic boundary conditions, energy would be allowed to spuriously re-enter the system, which is obviously non-physical. A usual method to circumvent this problem is to impose artificial absorbing boundaries. This can be done by making $V$ imaginary [17], thereby eliminating the energy that reaches the border of the computational grid. This option has been incorporated in the BPM software library.

The BPM software library is useful to model a wide range of physical systems governed by equation (1). Some examples will be presented in section 3.

2.2. The BPM library and examples

The BPM library consists of a set of pure Python functions, with explicit calls to numerically efficient routines of NumPy. When designing the library, advanced programming techniques, such as object oriented design patterns, have been avoided in order to make the algorithms easier to understand. Apart from the simple design, the BPM library offers several advantages for use by educators and students: it is open source; it is based upon the popular general purpose script language Python, and it can be easily adapted to other similar physical systems based on the Schrödinger equation. The fact that it is easy to read makes it ideal for beginners [18] to focus on the essential aspects of the numerical algorithms, and not on arcane programming details. Moreover, Python is an active language that is popular amongst scientists [19], having ample online documentation and technical support forums. Amongst the other advantages of Python’s appeal is that it is cross-compatible and has a large collection of numerical, symbolic, and visualisation packages.

In the supplementary material [4], the main Python BPM library code is provided. Detailed user instructions of the code and examples are also given, together with an installation tutorial in Windows, Linux, and Mac operating systems. Links to step-by-step video tutorial videos can also be found in instruction notes [4].

3. Examples

Together with the BPM library package, 20 example scripts—representing different physical systems of interest in 1D and 2D—are provided (see supplementary material [4]). These examples cover numerical approximations of textbook formulae as well as computations.
related to recent research in NLWs. Here, each example is described in the context of the
BPM library and with respect to its dimensionality and linearity.

3.1. Linear examples in 1 + 1D

A practical computational application of the BPM library is the quantum scattering process in
one spatial dimension. An interesting example is when the potential takes the form
\[ V = -\frac{s(s + 1)}{\cosh^2 x}. \]  
This is a well-known case for which the potential becomes reflectionless when \( s \) is a positive
integer. Figure 1 shows simulation results using the BPM library illustrating this curious
phenomenon. An interesting exercise for students is to explore results for different integer and
non-integer values of \( s \), verifying the observations of [20].

Apart from the quantum scattering example of figure 1, four other linear 1 + 1D
examples are included with the library. In particular, the cases treated are beam diffraction by
a slit (Diffraction_Slit_1D), interference pattern generation when two coherent beams
meet (Interference_Gaussians_1D), wave function beating within a double potential
well in relation to quantum tunnelling (Double_Well_1D), and a scattering process from a
rectangular barrier (Rectangular_Barrier_1D).

3.2. Nonlinear examples in 1 + 1D

The cubic potential
\[ V = \kappa |\psi|^2 \psi \]  
is used to describe several phenomena in physics, and has been studied extensively from
mathematical and computational perspectives [21, 22]. In nonlinear optics, the potential
corresponds to a Kerr term, while for ultracold Bose–Einstein condensates it describes a two-
body local atomic interactions. When \( \kappa = -1 \), the interaction is attractive and results in a
stable bright soliton [23],
\[ \psi = \exp \left[ \frac{i}{2} \left( t + 2vx - v^2 t + \phi \right) \right] \text{sech}(x - x_0 - vt), \]  
where \( x_0 \) is the initial position, \( v \) is the velocity, and \( \phi \) is a constant phase. Figure 2 shows the
simulation results of the interaction between two solitons having different initial relative
phases. As depicted in the figure, the self-trapped solitons propagate undisturbed outside the
interaction region, emerging unchanged from the collision.

Also included as computational examples in the BPM library are the following physical
systems: (a) soliton emission from a shallow atomic trap with space-dependent [24]
(Soliton_Emission_A_1D), (b) time-dependent [25] (Soliton_Emission_B_1D)
nonlinearities, and (c) a system in which the Thomas–Fermi approximation is initially valid
(Thomas_Fermi_1D) (i.e. the Laplacian term is ignored for finding an analytical approx-
imation of the ground state). For this latter case, the strength at time \( t \) of the nonlinearity is
changed and the wave envelope starts oscillating.

3.3. Linear examples in 1 + 2D

The linear two-dimensional Schrödinger equation provides a useful model for quantum
mechanical description of light beam propagation [5]. In this context, the time variable \( t \) is
replaced by propagation distance. Analytic solutions of equation (1) with vanishingly small
\[ V \text{ is given by } \psi = \frac{C}{(t + i \tau R)^{l+1}} r' e^{\frac{i r^2}{2(t + i \tau R)}} e^{\pm il \phi}. \]

where \( C, \tau R \) are real constants, \((r, \phi)\) are polar coordinates for the \((x, y)\) plane, and \( l = \ldots, -2, -1, 0, 1, 2, \ldots \) is the topological charge. The Gaussian beam corresponds to Figure 1. Reflectionless scattering of a Gaussian wave packet from the potential of equation (3) with \( s = 10 \). The sequential panels (a)–(e) represent \(|\psi|^2\) for a series of growing values of \( t \). The figure corresponds to the example [4] Sech2_Pot_1D.
while the expressions with \( l = 0 \) are vortex solutions with a phase singularity at their centre; such beam profiles have been extensively studied in optics. Figure 3 shows the resulting interference pattern of two overlapping beams with \( l = 0 \) and an \( l = 1 \). The characteristic fork-like structure indicates the order and sign of the phase dislocation [26].

Diffraction patterns from the interaction of vortices with topological charge \( l \) can be observed experimentally by using a computer-generated hologram having a fork-like form (i.e. the Y-like structures) [26]. A simulation of this process is displayed in figure 4.

In the BPM library, there are two other example cases that compute the behaviour of beams described by equation (6) with \( l = 0 \) (Gaussian Beam 2D) and \( l = 1 \) (Vortex 2D). Also related to beam propagation, another example code describes the diffraction by a circular hole (Diffraction Circle 2D).

3.4. Nonlinear examples in 1 + 2D

The nonlinear Schrödinger equation in 1 + 2 dimensions is an important problem in nonlinear optics, resulting in countless contributions in the scientific [22] and educational literature [5, 27]. An example of the latter in [14] describes a graphical user interface for helping students explore the nonlinear Schrödinger equation. Among the many interesting nonlinear phenomenological studies reported is filamentation [28], where a wide beam with a Kerr-like
focusing nonlinearity equation (4), with $\kappa < 0$, becomes unstable due to modulational instability; collapse is impeded due to other processes and light filaments are formed. By using a simple model, this phenomenon can be simulated; figure 5 shows the case where optical collapse is prevented by including a higher order defocusing nonlinearity, as required in optical filamentation.

The dynamics of Bose–Einstein condensation (BEC) of (trapped) atoms, are also described by the nonlinear Schrödinger equation; in this context, it is referred to as the Gross–Pitaevskii equation. With an asymmetric trapping potential that is much stronger along one direction, the atom cloud is flattened, so that the dynamics take place in a two-dimensional

![Figure 4.](image)

**Figure 4.** The simulated evolution of an initially Gaussian beam passing through a holographic mask. The resulting diffraction pattern consists of a set of vortices and antivortices with different topological orders. The results were generated using the example code [4] Vortices_Pattern_2D from the BPM library.

![Figure 5.](image)

**Figure 5.** A process resembling optical filamentation. The initial condition is a constant wave function perturbed by random noise. Inhomogeneities grow because of the nonlinear interaction, eventually producing a paradigmatic pattern. The simulation corresponds to the example Python script Filamentation_2D included in the BPM library [4].
plane. Figure 6 shows a simulation (using the example code Vortex_Preseccion_2D in the BPM library) for a BEC system recently described in the literature [29]. For a linear harmonic potential, a vortex displaced from the centre has a periodic orbit. By introducing a weak nonlinearity, the vortex undergoes precession and the evolution of the phase singularity is not circular. In [29], analytic expressions were derived to first order in the nonlinear perturbation.

Further nonlinear phenomena are treated in other Python scripts included with the BPM library. With attractive nonlinearities, vortices may become unstable with respect to azimuthal perturbations, causing the $|\psi|^2$ distribution to eventually break apart; the example code, (Vortex_Breaking_2D), treats this case. Potentials with a cubic-quintic (attractive-repulsive) nonlinearity [30] have been widely studied, since light beams in such media behave like liquids [31]; the example code Liquid_Droplet_2D describes a light droplet that collides with a potential barrier. Finally, self-similar wave function collapse with Townes profile is treated in the example code Collapse_2D (that also which requires townes_profile.csv to define the initial condition).

4. Scope, limitations and possible future developments of the software library

Because the main purpose of the BPM software library is educational, it was designed to be simple and widely applicable, yet have sufficient computational performance. In this way, the library functions can be used to model other physical 1D and 2D situations that obey equation (1) without modifying the functions themselves, but simply specifying the potential, boundary, and initial conditions. A simple extension to the existing functions would be to include one or more extra wave functions. In optics, two coupled wave functions could correspond to beams of different polarisations or wavelengths. In Bose–Einstein condensates, they could correspond to different atomic species or hyperfine levels. For some physical
systems, the potential $V$ is defined non-locally in terms of $|\psi|^2$ (e.g. [9, 10]), a property that can be incorporated in the code. Finally, the BPM software library calculates (real) time propagation of the wave function, but does not find the eigenstates of the Schrödinger equations. However, the ground states could be found by propagating in imaginary time [32], which could be implemented with minimal changes to the BPM library code. Another obvious extension is the generalisation to three dimensions; in this case, considerable extra effort would be required for visualisation of the results.

Nonetheless, the intentional simplicity of the BPM software library introduces obvious limitations. Many cases described involving the integration of equation (1) can be accomplished with more suitable and/or powerful algorithms, adapted to the problem at hand. For instance, some examples correspond to free propagation ($V = 0$), not requiring the split-step procedure. Apart from obvious cases, the split-step procedure can be optimised depending on the particular features of the problem [33]. Finally, it is often the case that adaptive grids are used in practice to accommodate largely different spatial scales.

With respect to numerical accuracy, the split-step method as described here is of first order; i.e. the numerical error scales as $O(dr)$ (the error at each step is of order $dr^2$ and the number of steps is proportional to $dr^{-1}$). Higher order split-step methods are treated in the literature [34] and should be used for cases where high precision is necessary. Finally, the split-step method is amenable to optimisation on parallel computing and/or general purpose computing on graphics processing, which are capable of achieving dramatic speed improvements, especially when the spatial grid is large.

5. Suggestions for the utilisation of the software with educational purposes

Inquiry-based science teaching—in which the student can actively experiment and draw conclusions—has been shown to achieve superior outcomes [35]. With this teaching methodology, students learn by engaging in scientific observations. The virtues of computer simulation are now well established. A virtual laboratory, where computational tools are packaged for highly specific domains and for elucidating a particular phenomenon, are advantageous teaching concepts that are difficult to understand and to programme by novice students. Such virtual laboratories can also broaden a novice student’s exposure to contemporary issues in physics, while offsetting the high costs of actual experimental laboratories [36]. The BPM software library fulfills this didactic objective for exploring the specific domain of Schrödinger dynamics in different physical situations; making it easy to exchange different potentials, boundary conditions, visualise the results, and easily explore the numerical methods under the hood. By exposing the potential and boundaries, while encapsulating the integration details in the library, the student can rapidly start focusing on the physics. With this intention, the code has a modular design, without the need to modify core functions, while addressing a large class of physical configurations.

Conversely, advanced students interested in the practical implementation of the numerical algorithms will find the main routines easy to read and understand. This code serves to reduce the gap between physics coursework and pseudospectral methods implemented in the scientific research literature.

Here, we suggest a tentative plan for guiding undergraduate students in making computations with the BPM library in subjects such as quantum mechanics, optics, or aspects of NLWs. Graduate students should be able to advance in their training by reading this text and working through the computer codes with minimal guidance. In any case, the advice of
teachers and supervisors is of fundamental importance to select the relevant problems to address.

(i) In an introductory computer laboratory, the instructor could provide an installed and executable version of the library, so students avoid any details of the install process. The instructor could choose between one or more linear or nonlinear examples to illustrate particular topics related to the objectives of the course. The students would use the examples of the BPM library to observe the resulting evolution of the wave function due to modifications of set of physical parameters (e.g. initial width and velocity of a wave packet).

(ii) After understanding the results of the example problem, students could define a new or similar problem, not just by tuning parameters but by generating new systems having different forms of the potential and initial conditions. Ideally, they should be encouraged to reproduce results from textbooks or recently published research.

(iii) Using the instructions provided in the library, the students could install the software on their own computers in order to work independently outside the classroom.

(iv) The instructor or student could also make modifications to the main structure of the code in order to optimise it for whatever concrete problem is under study, or to improve the visualisation of the results.

For those interested in algorithmic aspects of computational physics, the following activities could be of interest:

(a) Modify the computational parameters such as grid size to study its dependence on computational time and precision.

(b) Analyse the code of the BPM library in detail and implement general or specific changes that could improve computational performance for a particular case. For this, the comments in the scope section 4 could be of interest.

(c) Create a similar software library for a different kind of problem and/or with a different computational algorithm. Take advantage of the available literature, e.g. [12, 18, 19].

We close the section by suggesting a number of exercises, in connection to the previous points, that may provide additional ideas for instructors and students for developing a learning plan with the BPM library.

• Explore the scattering of wave packets from barriers, dips, or steps of different forms and strengths. Implement the numerical integrals to compute the reflection and transmission coefficients.

• Reproduce the analytic results of [37] using numerical computations and test the precision of the algorithm for different computational parameters.

• Study the scattering of a bright soliton from a potential barrier. Implement and study the dynamics of dark solitons as described in [8, 38].

• Reproduce the results found in [39] for a cubic-quintic nonlinearity.

• Compare the numerical results of the linear evolution of a gaussian beam and a vortex beam to the analytical expression of equation (6). Study the dependence of the solution with respect to different time step \( dt \) and spatial grid sizes.

• Check the marginal stability of the Townes profile by considering small modifications of the strength of the cubic nonlinearity.

• Study the generation of rarefaction pulses when bright pulses collide in a cubic-quintic medium [40].
6. Conclusions

We described an educational software library for solving and exploring the time-dependent Schrödinger equation. For a course in numerical and computational physics, this software could be used as a virtual laboratory—simulators for a highly specific phenomenological domain—to simulate quantum mechanics, beam optics and NLVs. The BPM software library is open source and purposely designed to be simple so as to be used on any computer, with low computational demands. The study of the methods and the software could be useful for students getting started in computational physics and in the Python programming language. Example scripts included with the BPM library cover a wide range of physical 1D and 2D physical systems illustrating fundamental concepts useful to prepare the young researcher for topics in modern research topics.

The opportunities for understanding physical systems and predicting compelling phenomena with different versions of the Schrödinger equation are inexhaustible. The BPM library should be a useful starting point for undergraduate or graduate students interested in this exciting topic.

Acknowledgments

We thank A F Biasi, J Blanco-Labrador and D Tommasini for testing the code and commenting on the manuscript. This work is supported by grants FIS2014-58117-P and FIS2017-83762-P from Ministerio de Economía y Competitividad (Spain), and grant GPC2015/019 from Consellería de Cultura, Educación e Ordenación Universitaria (Xunta de Galicia).

ORCID iDs

David Olivieri © https://orcid.org/0000-0001-7862-6917
Angel Paredes © https://orcid.org/0000-0003-3207-1586
Humberto Michinel © https://orcid.org/0000-0002-7854-7626

References

[1] De Jong T, Linn M C and Zacharia Z C 2013 Physical and virtual laboratories in science and engineering education Science 340 305–8
[2] Zacharia Z C, Olympiou G and Papaevripidou M 2008 Effects of experimenting with physical and virtual manipulatives on students’ conceptual understanding in heat and temperature J. Res. Sci. Teach. 45 1021–35
[3] Hatherly P A, Jordan S E and Cayless A 2009 Interactive screen experiments-innovative virtual laboratories for distance learners Eur. J. Phys. 30 751–62
[4] The software library and its documentation can be downloaded from: https://github.com/pyNLSE/bpm
[5] Teich M C and Saleh B E A 2007 Fundamentals of Photonics 2nd edn (New York: Wiley)
[6] Hasegawa A and Tappert F 1973 Transmission of stationary nonlinear optical pulses in dispersive dielectric fibers: I. Anomalous dispersion Appl. Phys. Lett. 23 142–4
[7] Davydov A S 1979 Solitons in molecular systems Phys. Scr. 20 387–94
[8] Burger S, Bongs D, Dettmer S, Ertmer W, Sengstock K, Sanpera A, Shlyapnikov G and Lewenstein M 1999 Dark solitons in Bose–Einstein condensates Phys. Rev. Lett. 83 5198–201
[9] Paredes A and Michinel H 2016 Interference of dark matter solitons and galactic offsets Phys. Dark Univ. 12 50–5
[10] Briscese F 2017 The Schrödinger–Poisson equations as the large-N limit of the Newtonian N-body system: applications to the large scale dark matter dynamics Eur. Phys. J. C 77 623
[11] Schmidt J D 2010 Numerical Simulation of Optical Wave Propagation with Examples in MATLAB (Bellingham, WA: SPIE)
[12] Gould H, Tobocznik J and Christian W 2016 An Introduction to Computer Simulation Methods (New York: Addison-Wesley)
[13] Orquin I, Garcí a-March M A, Fernandez de Córdoba P, Urchegui A J F and Monsoriu J A 2007 Introductory quantum physics courses using a LabVIEW multimedia module Compt. Appl. Eng. Educ. 15 124–35
[14] Cambroner-López F, Gómez-Varela A and Bao-Varela C 2017 Designing an ultrafast laser virtual laboratory using Matlab Guide Eur. J. Phys. 38 034006
[15] Chhabra M and Das R 2016 Quantum mechanical wavefunction: visualization at undergraduate level Eur. J. Phys. 38 015404
[16] Fleck J Jr, Morris J and Feit M 1976 Time-dependent propagation of high energy laser beams through the atmosphere Appl. Phys. 10 129–60
[17] Agrawal G P 2007 Nonlinear Fiber Optics 4th edn (Burlington, VA: Academic)
[18] Landau R H, Páez M J and Bordeianu C C 2015 Computational Physics: Problem Solving with Python 3rd edn (New York: Wiley)
[19] Newman M 2013 Computational Physics (North Charleston, SC: CreateSpace Independent Publ.)
[20] Kiriuščėnienė N and Kuzmin S 1998 Scattering of a Gaussian wave packet by a reflectionless potential Am. J. Phys. 66 867–72
[21] Sulem C and Sulem P-L 1999 The Nonlinear Schrödinger equation: Self-Focusing and Wave Collapse (New York: Springer)
[22] Fibich G 2015 The Nonlinear Schrödinger equation (Cham: Springer)
[23] Zakharov V E and Shabat A B 1972 Exact theory of two-dimensional self-focussing and one-dimensional self-modulating waves in nonlinear media Sov. Phys.—JETP 34 62–9 http://www.jetp.ac.ru/cgi-bin/e/index/e/34/1/p62?a=list
[24] Rodas-Verde M I, Michinel H and Pérez-García V M 2005 Controllable soliton emission from a Bose–Einstein condensate Phys. Rev. Lett. 95 153903
[25] Michinel H, Paredes A, Valado M M and Feijoó D 2012 Coherent emission of atomic soliton pairs by Feshbach-resonance tuning Phys. Rev. A 86 013620
[26] Carpentier A V, Michinel H, Salgueiro J R and Olivieri D 2008 Making optical vortices with computer-generated holograms Am. J. Phys. 76 916–21
[27] Rogel-Salazar J 2013 The Gross–Pitaevskii equation and Bose–Einstein condensates Eur. J. Phys. 34 247
[28] Couairon A and Mysoyrowicz A 2007 Femtosecond filamentation in transparent media Phys. Rep. 441 47–189
[29] Biasi A, Bizon P, Craps B and Evnin O 2017 Exact lowest-Landau-level solutions for vortex precession in Bose–Einstein condensates Phys. Rev. A 96 053615
[30] Wu W, Zhang Y, Yuan C, Wen F, Zheng H, Zhang Y and Xiao M 2013 Cubic-quintic condensate solitons in four-wave mixing Phys. Rev. A 88 063828
[31] Michinel H, Campo-Táboas J, García-Fernández R, Salgueiro J R and Quiroga-Teixeiro M 2002 Liquid light condensates Phys. Rev. E 65 066604
[32] Lehtovaara L, Toivanen J and Eloranta J 2007 Solution of time-independent Schrödinger equation by the imaginary time propagation method J. Comput. Phys. 221 148–57
[33] Shao J, Liang X and Kumar S 2014 Comparison of split-step Fourier schemes for simulating fiber optic communication systems IEEE Photonics J. 6 1–15
[34] Muslu G M and Erbay H A 2005 Higher-order split-step Fourier schemes for the generalized nonlinear Schrödinger equation Math. Comput. Simul. 67 581–95
[35] Minner D D, Levy A J and Century J 2010 Inquiry-based science instruction—what is it and does it matter? Results from a research synthesis years 1984–2002 J. Res. Sci. Teach. 47 474–96
[36] Galan D, Heradio R, de la Torre L, Dormido S and Esquembre F 2017 The experiment editor: supporting inquiry-based learning with virtual labs Eur. J. Phys. 38 035702
[37] van Dijk W, Toyama F M, Prins S J and Spyksma K 2014 Analytic time-dependent solutions of the one-dimensional Schrödinger equation Am. J. Phys. 82 955–61
[38] Kivshar Y S and Luther-Davies B 1998 Dark optical solitons: physics and applications Phys. Rep. 298 81–197
[39] Kim W-S and Moon H-T 2000 Soliton-kink interactions in a generalized nonlinear Schrödinger system Phys. Lett. A 266 364–9
[40] Paredes A, Feijoó D and Michinel H 2014 Coherent cavitation in the liquid of light Phys. Rev. Lett. 112 173901