

## Physlets and Open Source Physics for Quantum Mechanics: Visualizing Quantum-mechanical Revivals

**Mario Belloni**

Assistant Professor of Physics

[mabelloni@davidson.edu](mailto:mabelloni@ davidson.edu)

**Wolfgang Christian**

Professor of Physics

[wochristian@davidson.edu](mailto:wochristian@davidson.edu)

*Davidson College  
Davidson, NC 28035*

### Abstract

*In this paper we describe our five-year effort to create interactive curricular material for upper-level quantum mechanics courses. This material uses both Physlets and newly created Open Source Physics applets and applications to make the teaching of quantum mechanics visual and interactive. These exercises and tools address both quantitative and conceptual difficulties experienced by many students. Because the materials are Web based, they are extremely flexible and are appropriate for use with various pedagogies, such as the Just-in-Time Teaching technique. We briefly outline the features of Physlets and Open Source Physics programs and then describe our suite of Java programs that solve and visualize the problem of a wave packet in an infinite square well. The materials described in this paper can be found on the Open Source Physics Web site and on the MERLOT and ComPADRE digital libraries.*

### Introduction

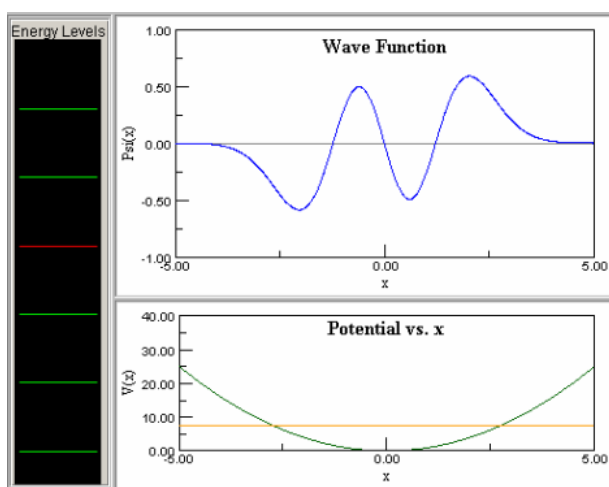
On the surface, the teaching of quantum mechanics seems like a specialized topic for those teaching physics. In fact, because of its fundamental role in physics and chemistry, quantum mechanics is one of the most widely taught topics on the college and university level. Undergraduates encounter quantum mechanics in their introductory physics, modern physics, physical chemistry, statistical and thermal physics, and quantum mechanics courses, while graduate students may take as many as five additional courses in quantum mechanics and quantum field theory. Despite the importance of quantum theory, the teaching of quantum mechanics in chemistry and physics has not changed significantly since quantum mechanics was invented. In addition, studies have shown that there is very little difference between an

undergraduate's conceptual understanding of quantum mechanics and a graduate student's conceptual understanding of quantum mechanics [1]. Recent advances in computational power, however, have made many heretofore inaccessible topics in theoretical and computational quantum mechanics more accessible to teachers and their students. We have created many exercises for studying energy eigenstates (so-called stationary states) which have been reported elsewhere [2,3]. This paper describes the technology and pedagogy of a suite of Java programs that explore the dynamics of quantum-mechanical wave packets in an infinite square well.

## Computer-based Pedagogy

### *Physlets*

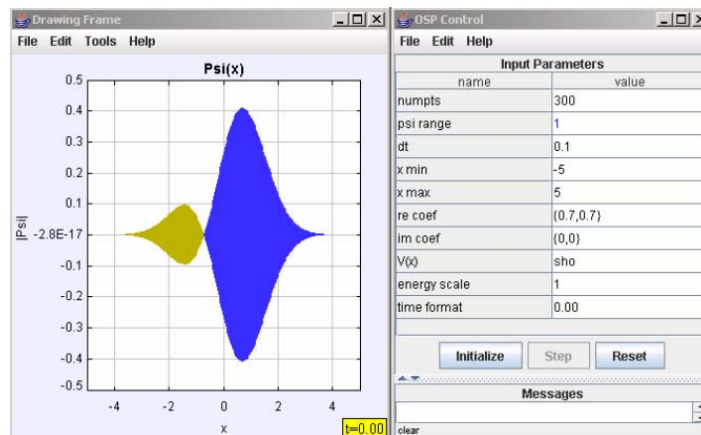
By now it is hard to imagine a teacher who has not heard the call to 'teach with technology' as it has resounded through educational institutions and government agencies alike. However, teaching with technology, just for the sake of teaching with technology, has often resulted in the development and use of tools that are not pedagogically sound [4,5]. Into this environment, we introduced Physlets®, a collection of interactive computer simulations [6] and accompanying curricular materials [7-11] designed with a sound use of pedagogy in mind [12-15]. The aim of Physlets is to provide a resource for teaching that enhances student learning through interactive engagement [16-18]. At the same time, Physlets are flexible enough (all Physlets can be set up and controlled with JavaScript) to be adapted to a variety of pedagogical strategies and local environments. Physlets are free for noncommercial use and are based on HTML and open internet standards (Physlets run on any platform: Macintosh OSX Panther, Windows, and Linux, with Java-enabled browsers that support Java-to-JavaScript communication such as: Safari, Internet Explorer, or Mozilla). Physlet-based curricular material is easy to translate into other languages and Physlet Web sites can be found throughout the world [19]. While we have focused on topics from physics, such as quantum mechanics, Physlets can be used to create exercises for many other fields such as mathematics, chemistry [11], biology, and economics [2].



**Figure 1:** An exercise from time-independent quantum mechanics showing the flexibility of the Physlet paradigm. Shown are the wave function for the  $n = 3$  energy eigenstate, the energy spectrum (with the red line showing the current state), and the energy diagram.

The Physlet paradigm, an example of which is shown in Figure 1, is based on the Model-View-Control (MVC) design pattern which is one of the most successful software architectures ever invented. The model maintains data and provides methods by which that data can change; the control gives the user the ability to interact with the model using the keyboard and the mouse; and the view presents a visual representation of the model's data. Java is ideal for the implementation of models and views because these objects are often complex and computational speed is important. But since the control object is accessed infrequently and should be customizable, it can be implemented differently. A scripting language such as JavaScript embedded into an HTML document provides an ideal solution. The advantage of using JavaScript and HTML to control a model and its views is that a common user interface can be created to ease the learning curve across pedagogical contexts. This unique approach has contributed to the success and wide adoption of Physlets. For example, a Google search on "Physlets" returns over 12,000 Web pages.

### Open Source Physics



**Figure 2:** An OSP program, *QMSuperpositionApp*, used to depict a two-state superposition for the quantum-mechanical harmonic oscillator. The visualization (the view) is shown on the left while the OSP control is shown on the right. One can edit animation parameters by typing in the parameter fields. These parameter values can also be saved and loaded using an XML file as shown in Figure 3.

While Physlets are in many ways 'open,' they are not open source. As a consequence, the underlying Java code is inaccessible. This approach works for general-purpose applications, but fails for more sophisticated one-of-a-kind simulations that require advanced discipline-specific expertise, such as those for quantum mechanics. Users and developers of these types of programs often have specialized curricular needs that can only be addressed by having access to the source code. However, anyone who has ever written a program in Java knows that writing code for creating buttons, text fields, and graphs can be tedious and time consuming. The Open Source Physics (OSP) project [20] solves this problem by providing a consistent object-oriented library of Java components (OSP libraries are based on Swing and Java 1.4) for anyone wishing to write their own programs.

Open Source Physics curricular material is written in Java and distributed using internet technologies. Although we distribute source code under the GNU GPL license, the

modification and redistribution of compiled Java programs is not an easy task for many teachers and students. It is not, however, necessary to become expert in programming to use our materials. Although the source code will be available for Java experts, our programs store their internal parameters using Extensible Markup Language (XML). A teacher can, for example, load a pre-defined multi-state quantum superposition demonstration using the data saved as an XML file shown in Figure 3. Because XML data files are character-based, it is straightforward to edit properties such as the expansion coefficients to create custom simulations.

```
<?xml version="1.0" encoding="UTF-8" ?>
<object class="org.opensourcephysics.controls.OSPApplication">
<property name="control" type="object">
<object class="org.opensourcephysics.controls.AnimationControl">
  <property name="initialize_mode" type="boolean">>false</property>
  <property name="dt" type="string">0.01</property>
  <property name="x min" type="string">-5</property>
  <property name="x max" type="string">5</property>
  <property name="psi range" type="string">1</property>
  <property name="re coef" type="string">{0.7,0.7}</property>
  <property name="im coef" type="string">{0,0}</property>
  <property name="V(x)" type="string">sho</property>
</object>
</property>
<property name="model" type="object">
<object
class="org.opensourcephysics.davidson.qm.QMSuperpositionApp" />
</property>
</object>
```

Figure 3: XML code for the *QMSuperpositionApp* from Figure 2.

### Launcher

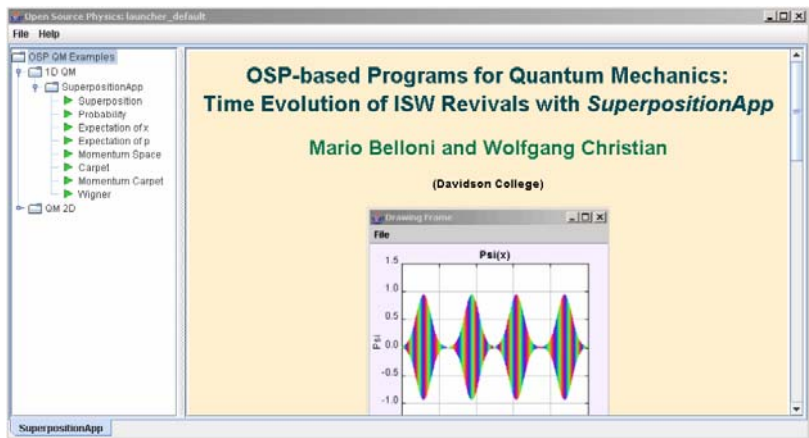


Figure 4: A *Launcher* window for the suite of quantum-mechanical revival simulations based on the program *QMSuperpositionApp*. Double clicking on *SuperpositionApp* opens a set of programs based on the *SuperpositionApp* program. Double clicking on an end-node executes the program.

Like Physlets, OSP provides a framework for embedding programs into HTML pages and for controlling these programs using JavaScript. Unlike Physlets, programs can

also be run as Java applications. The *Launcher* program shown in Figure 4 enables authors to distribute collections of OSP programs and curricular material in consistent self-contained packages. *Launcher* is designed to execute multiple Java applications or multiple copies of a single Java application with different initial conditions. *Launcher* also uses an XML file to store its properties, as shown in Figure 5.

```
<?xml version="1.0" encoding="UTF-8" ?>
<object class="org.opensourcephysics.tools.Launcher$LaunchSet">
  <property name="classpath" type="string">demo200407.jar</property>
  <property name="width" type="int">977</property>
  <property name="height" type="int">650</property>
  <property name="divider" type="int">220</property>
  <property name="launch_nodes" type="collection"
class="java.util.ArrayList">
  <property name="item" type="string">SuperpositionApp.xml</property>
  <property name="item" type="string">Quantum2D.xml</property>
</property>
</object>
```

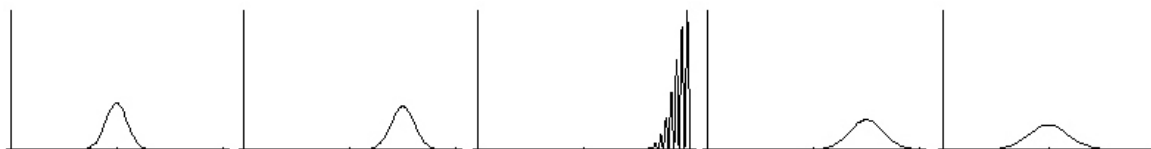
**Figure 5:** XML code for configuring the *Launcher* window shown in Figure 4. The size of the *Launcher* window is set and the *SuperpositionApp.xml* and *Quantum2D.xml* files are loaded. The XML files contain the 'outline' information shown in the left-hand panel of Figure 4.

Java-based curricular material has traditionally been distributed over the Internet using applets. Sun Microsystems has recently released a new technology known as Java Web Start that combines the applet and application approaches. Java Web Start is a helper application that is associated with a Web browser. When a user clicks on a link within an HTML page that points to a special file on the server, it causes the browser to launch Java Web Start, which then automatically downloads, caches, and runs the Java-based application. The entire process is typically completed without requiring any user interaction, except for the initial single click. A number of Java Web Start Launcher packages can be downloaded and run from the OSP Web site.

## Quantum-mechanical Wave Packets

### Wave Packet Revivals: Visualization

Time-dependent quantum-mechanical wave functions are inherently complex (having real and imaginary components) due to the time evolution governed by the Schrödinger equation. The standard way to visualize the wave function considers the probability density in position or momentum space (more typically position space), an approach that discards all phase information.



**Figure 6:** A free particle wave packet, initially moving to the right encounters an infinite wall on the right before reflecting and moving to the left. The image depicts the packet's time evolution at equal-time intervals. The middle image corresponds to the 'bounce': the time a classical particle would have hit the wall.

Figure 6 shows the short-term time evolution of the probability density of a Gaussian wave packet in an infinite well with walls at  $x = 0$  and  $x = 1$ . The packet has an initial momentum to the right and the images are shown at equal-time intervals. We are supposed to imagine the motion of the packet by reading the images from left to right. We notice that as the packet moves to the right its leading edge encounters the infinite wall first and reflects to the left back towards the middle of the well.

There are several interesting features about this problem that are not depicted in the image, however. While we get the general sense of what happens, we lose a lot of the specific details by focusing on just one facet of the motion. This is especially apparent at the collision with the wall. At the classical collision time, the classical momentum is zero, yet in the quantum-mechanical case, the average momentum of the packet is negative [21,22]. On the much longer revival time scale we are interested in visualizing the fractional and exact quantum-mechanical revivals of the initial wave packet. One can again look at the position-space probability density, but there is no guarantee that by just looking we can recognize precisely when the revivals occur. In addition, showing just one image lacks the ability to show how the change of initial conditions affects the resulting dynamics.

Well-crafted computer simulations, however, can. The following OSP examples show the time evolution of wave packet dynamics in the infinite square well. We focus on the quantum ‘bounce’ and quantum-mechanical revivals and use the time dependence of the expectation values [23], so-called quantum carpets [24] and, in the future, the joint position-momentum quasi-probability density using the Wigner function [25] to uniquely depict wave packet evolution in time. A brief tutorial on wave packet phenomena and quantum-mechanical revivals [26-28] can be found in the Appendix.

#### *Wave Packet Revivals: Simulations*

For our initial Gaussian wave packets we use a variety of initial momentum values corresponding to the central values of  $n_0 = 0, 10, 20, 40,$  and  $80$  (where  $p_0 = n_0\pi/L$ ), so that the ratio of classical periodicity to spreading time varies (for  $n_0 = 40$ ,  $T_{cl}/t_0 = 10/\pi$ , and significant spreading can be seen even over half a classical period). For ease of calculation, we also set  $2m = L = \hbar = 1$ , while allowing the user to vary  $x_0$ ,  $p_0$ , and  $\Delta x_0 = b/\sqrt{2}$ . Using our simplifications, the revival time,  $T_{rev}$ , becomes:  $T_{rev} = 2n_0T_{cl} = 2/\pi$ . For the infinite square well, longer time scales are not present due to the purely quadratic dependence of the energy eigenvalues. The revival time scale can clearly be much larger than the classical period for wave packets characterized by  $n_0 \gg 1$  (giving  $T_{rev}/T_{cl} = 2n_0$ , which for  $n_0 = 40$  gives  $T_{rev}/T_{cl} = 80$ ).

The program we use to study revivals, *QMSuperposition*, inputs the expansion coefficients,  $c_n$ , of a general quantum-mechanical superposition shown in Eq. (A2). The program does so by separately inputting the real,  $\text{Re}[c_n]$ , and imaginary,  $\text{Im}[c_n]$  parts of the expansion coefficients,  $c_n$  (which were separately calculated in *Mathematica*). An XML file, such as the one shown in Figure 3, stores these expansion coefficients and other appropriate initial conditions for the program.

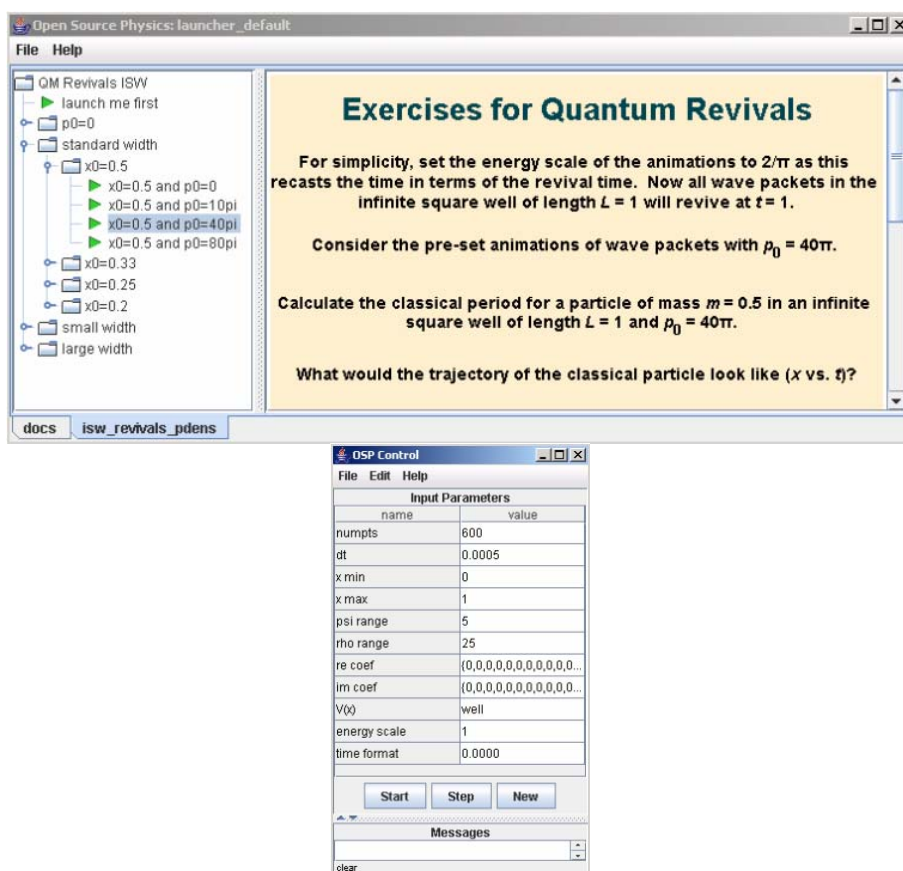
The wave functions are then calculated either numerically for any user-defined potential energy function,  $V(x)$ , or calculated analytically for the special cases of the infinite square well, the periodic infinite well, and the simple harmonic oscillator. Depending on the analysis one wishes to perform, one of the following programs based on *QMSuperpositionApp* (which itself only shows the wave function) can be chosen:

- *QMSuperpositionProbabilityApp*: adds a view of the probability density
- *QMSuperpositionExpectationXApp*: adds a view of the expectation value of  $x$

- *QMSuperpositionExpectationPApp*: adds a view of the expectation value of  $p$
- *QMSuperpositionCarpetApp*: adds a view of the position-space quantum carpet
- *QMSuperpositionMomentumCarpetApp*: adds a view of the  $p$ -space quantum carpet
- *QMSuperpositionFFTApp*: adds a view of the momentum-space wave function

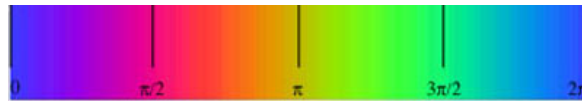
The use of analytic solutions for each eigenstate allows the simulation to run for a long period of time and yet never accumulate numerical error since the wave function is calculated anew at each time step. In addition, within the OSP control, we can set the energy scale, which determines the time scale for the animation. For example, setting the energy scale to  $2/\pi$  forces  $T_{\text{rev}} = 1$ , which is a convenient time scale for the study of revivals and fractional revivals which occur at  $T_{\text{rev}}$  and fractions of  $T_{\text{rev}}$ , respectively.

The applications described here can be organized using *Launcher*. Once *Launcher* is opened, the user can double click on a node to run a program with the initial conditions stored in the associated XML file.



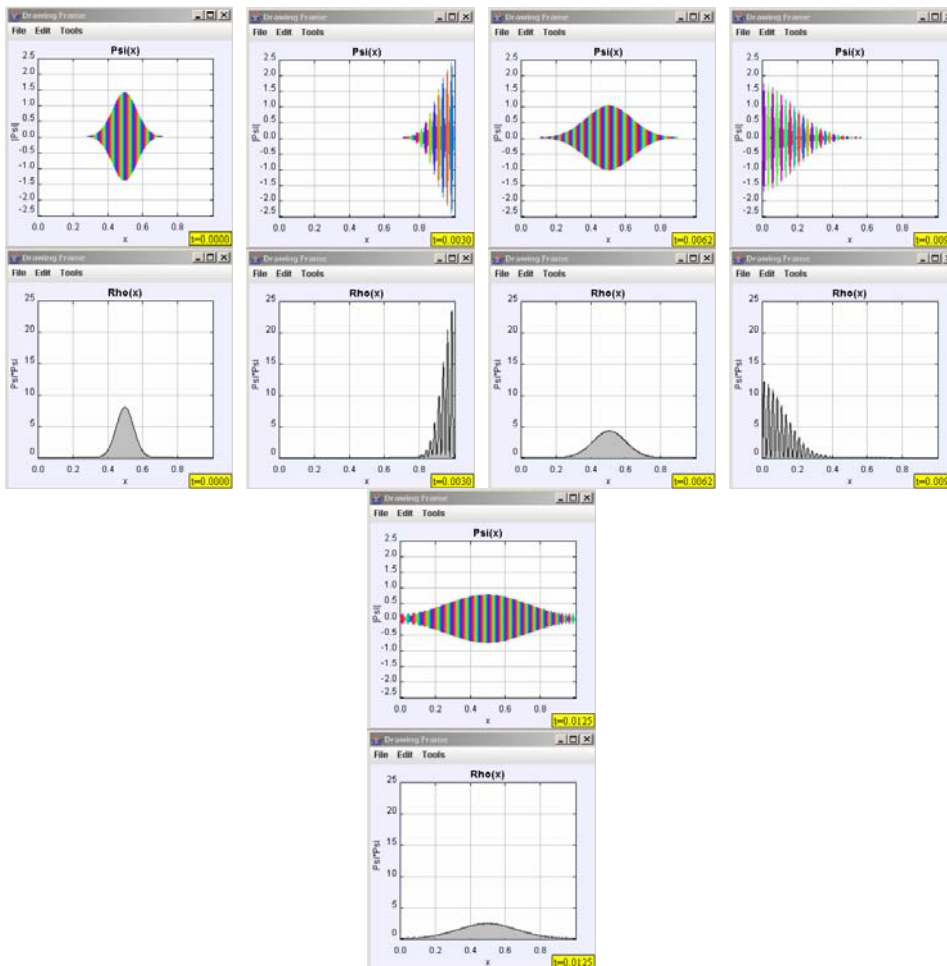
**Figure 7:** A *Launcher* window for a collection of quantum-mechanical revival simulations (on the left) based on the program *QMSuperpositionCarpetApp*. Selecting a given title for an animation by double-clicking opens the control window and loads XML properties for controlling the program (on the right). Once this control is open, one can edit animation parameters, read other XML files, or save a new set of parameters in an XML file.

## Evolution of a Quantum Wave Packet: Short- and Long-term Behavior in the Infinite Well



**Figure 8:** The color strip that serves as a conversion between the phase of the wave function and the color representation of the phase used in the animations.

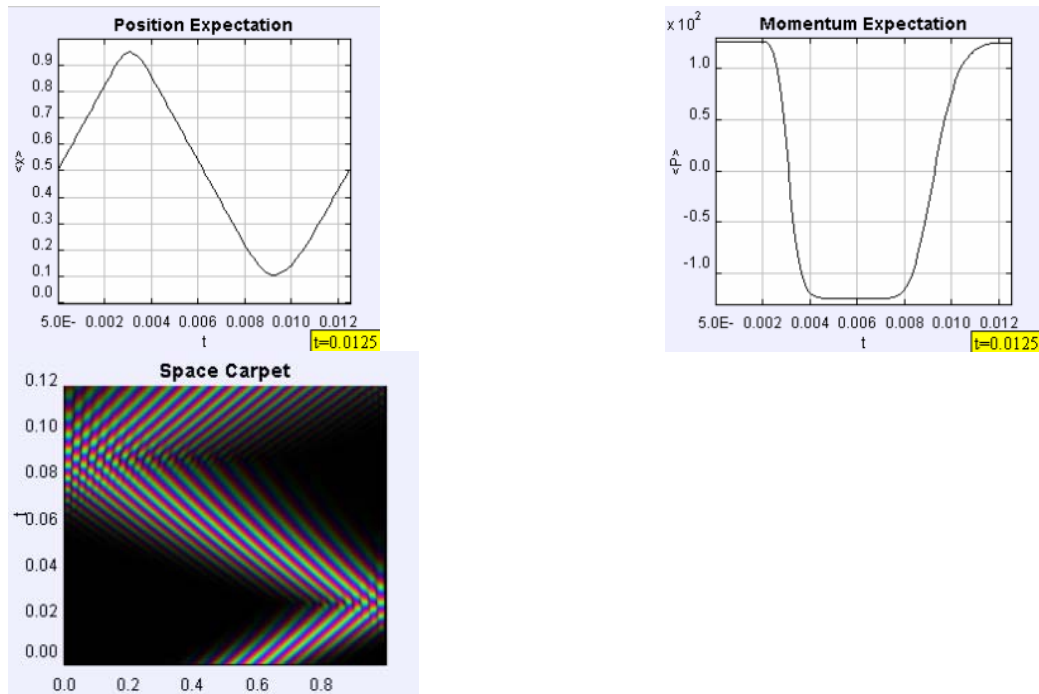
In the following examples we will focus on the time evolution of Gaussian wave packets in the infinite square well. In time-dependent quantum mechanics it is necessary to describe the wave function in terms of complex (real and imaginary) functions. In order to animate these time-dependent functions, we use an amplitude and phase representation of the wave function. The amplitude is drawn centered on the  $y$ -axis and is the height of the envelope shown from top to bottom. The phase, or phase angle, of the wave function is shown as color and we often show a color-conversion image, as in Figure 8 to help students determine the phase of the wave function.



**Figure 9:** A quantum wave packet in the infinite square well visualized with the program *QMSuperpositionProbabilityApp*. The wave packet is shown at  $t = 0$ ,  $T_{cl}/4$ ,  $T_{cl}/2$ ,  $3T_{cl}/4$ , and  $T_{cl}$ , respectively ( $T_{rev}$  is set to 1). During the classical period, the wave packet has begun to noticeably spread. In the top panels the wave function is shown in phase as color representation and in the bottom panels the probability density is shown.



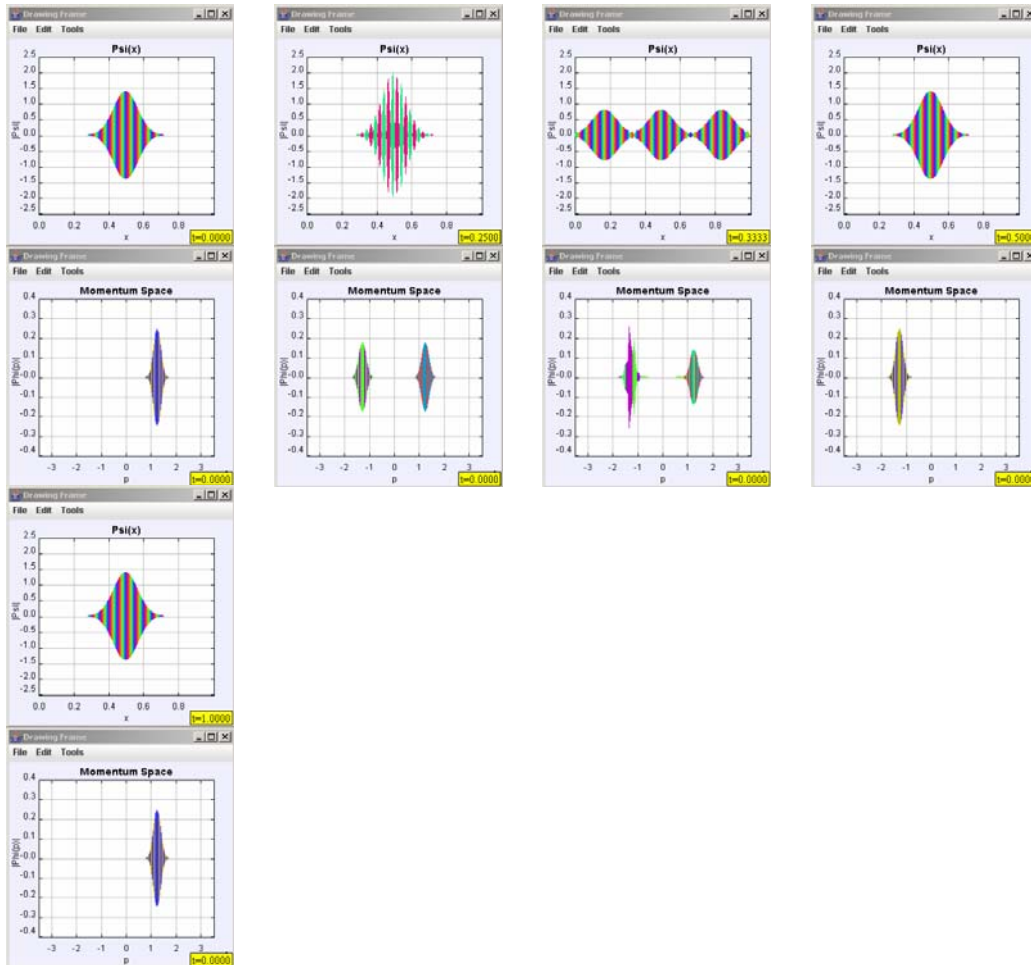
In Figure 9 we show the short-term dynamics for a Gaussian wave packet ( $x_0 = 0.5$ ,  $p_0 = 40\pi$ , and  $\Delta x_0 = 0.05$ ) in an infinite square well. As time goes on (from left to right in the figure), the wave packet moves to the right, bounces, moves to the left, bounces again, and moves to the right. At the same time the packet is spreading. Over one classical period the packet has already spread so much that it covers the entire extent of the well. Notice that at the times of  $T_{cl}/4$  and  $3T_{cl}/4$  the wave packet is colliding with one of the infinite walls. Unfortunately from this depiction, we cannot discern any of the interesting (and perhaps unexpected) features of the collision with the wall.



**Figure 10:** The time evolution of the same quantum wave packet in shown in Figure 9 between  $t = 0$  and  $t = T_{cl}$  ( $T_{rev}$  is set to 1) using three different OSP programs. Here we suppress the animation of the wave packet and show the expectation value of position (*QMSuperpositionExpectationXApp*), expectation value of momentum (*QMSuperpositionExpectationPApp*), and the quantum carpet (*QMSuperpositionCarpetApp*) visualizations instead.

In Figure 10, we are again showing the short-term classical time scale, but now with three different programs based on *QMSuperposition* (*\*ExpectationXApp*, *\*ExpectationPApp*, *\*CarpetApp*) that visualize the expectation value of position, the expectation value of momentum, and the quantum carpet. All three of these visualizations show the entire time evolution from  $t = 0$  and  $t = T_{cl}$ . Notice that the expectation value of position appears to be decreasing in amplitude over time. This effect is a direct result of packet spreading. In addition, there is a ‘softening’ of the collision with the wall as compared to what one expects classically. This ‘softening’ also occurs for the expectation value of the momentum, since it obeys Ehrenfest’s principle  $d\langle x \rangle/dt = \langle p \rangle/m$ . In addition, by zooming in on either the position or the momentum expectation values at the classical collision time,  $T_{cl}/4$ , one can see that the packet has a negative momentum in contradiction to the classically-expected result of zero. The position-space quantum carpet also shows the short-term time evolution of the packet, but it shows a space-time diagram (time vs. position) for the wave function. The colors represent the phase of the wave function and hence the lines of constant

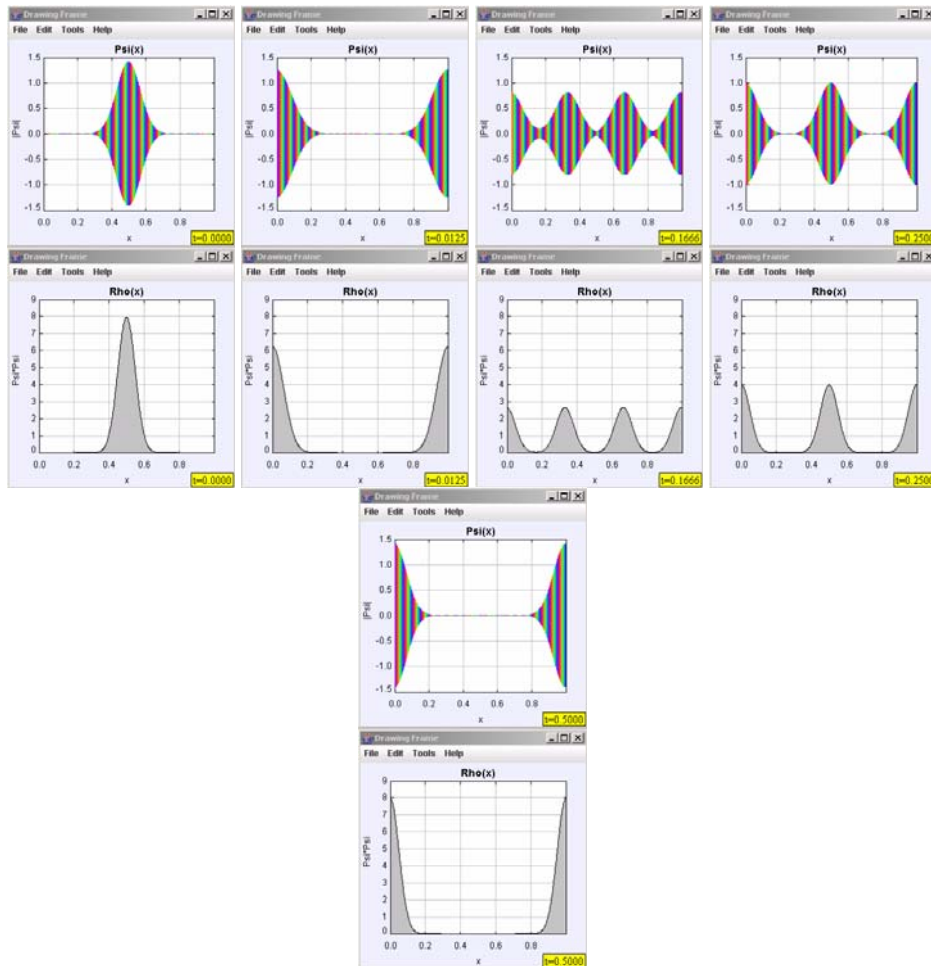
color represent the phase velocity of the wave packet. This phase velocity is exactly half that of the group velocity, which is a well-known quantum-mechanical result.



**Figure 11:** The position and momentum space time evolution of the same quantum wave packet is shown in Figure 9 now using the program *QMSuperpositionFFTApp*. The wave packet is shown at  $t = 0$ ,  $T_{\text{rev}}/4$ ,  $T_{\text{rev}}/3$ ,  $T_{\text{rev}}/2$ , and  $T_{\text{rev}}$  ( $T_{\text{rev}}$  is set to 1). During the revival time scale, the wave packet is usually spread out over the entire length of the well. The exceptions occur at fractions of the revival time:  $pT_{\text{rev}}/q$ , where  $p$  and  $q$  are integers. In the top panels the wave function is shown in phase as color representation and in the bottom panels the wave function is shown in momentum space. The momentum-space wave function is useful in discerning the multiple 'mini-packets' that arise during the fractional revivals.

On the revival time scale, as shown in Figure 11, the wave packet can reform into localized wave packets. At  $T_{\text{rev}}/4$ , two 'mini-packets' form at the exact same position, but with opposite momentum. This is clearly depicted with the momentum-space wave function, using the *QMSuperpositionFFTApp* program, in which two distinct momentum distributions are present. At  $T_{\text{rev}}/3$ , three mini-packets form. At half the revival time,  $t = T_{\text{rev}}/2$ , the wave packet also reforms, but at a location 'mirrored' about the center of the well. One can see from the momentum-space wave function that the packet is moving with 'mirror' (opposite) momentum values as well. Finally, at  $T_{\text{rev}}$ , the wave packet is exactly the same as when we started at  $t = 0$ .

## Time Evolution of a Quantum Wave Packet on a Ring



**Figure 12:** A free-particle Gaussian wave packet confined to a ring visualized with the same program used in Figure 9, but in this case the ‘ring’ is chosen for the potential instead of the infinite square ‘well.’ The wave packet is shown at  $t = 0, T_{cl}/4, T_{rev}/6, T_{rev}/4,$  and  $T_{rev}/2,$  respectively ( $T_{rev}$  is again set to 1, but for the ring, this is one quarter the revival time of the infinite square well). In the top panels the wave function is shown in phase as color representation and in the bottom panels the probability density is shown. The packet moves left to right and leaves the right-hand side to re-enter on the left-hand side.

The final example we will discuss is that of a Gaussian wave packet confined on a circular ring using the program *QMSuperpositionProbabilityApp* and the potential energy function,  $V(x)$  set equal to “ring.” This is the same program used in Figure 9, but now it uses periodic boundary condition. Here we visualize this example in two-dimensions, but elsewhere we have used the new Open Source Physics Java3D library [3,20] to show a three-dimensional depiction of the wave packet’s evolution. As shown in Figure 12, the Gaussian wave packet evolves over time and undergoes quantum-mechanical revivals. For this system, the wave packet again has an average momentum, causing it to move around the ring, and again it spreads. After a short time, the spreading is large enough that the front of the wave packet starts to interfere with the back of the wave packet. At fractions of  $T_{rev}$ , the revival time, the wave packet, or ‘mini-packet clones’ of the original wave packet, can form.

## **Conclusion**

We have created over 150 Physlet- and Open Source Physics-based exercises for the teaching and learning of quantum mechanics. Some of these exercises can be found on the Web site in Ref. [2]. Most of the new exercises, such as the ones presented in this paper, can be found on the Open Source Physics Web site (<http://www.opensourcephysics.org>) and on the MERLOT (<http://merlot.org>) and ComPADRE (<http://compadre.org/quantum>) digital libraries by searching for “Physlets” or “Open Source Physics.” These materials provide a new, exciting, and effective way to deliver interactive curricular material to students studying quantum mechanics. The effectiveness of these materials is supported by our preliminary assessment of the use of interactive curricular material in our quantum mechanics courses [2]. We are currently developing additional material that will appear in the book: *Physlet Quantum Mechanics* (2005 Prentice Hall).

## **Acknowledgements**

We would like to thank our colleagues at Davidson College for their support of this work. We would also like to thank Larry Cain and Bruce Mason for their helpful feedback in preparing this manuscript. This work has been supported in part by Associated Colleges of the South Teaching with Technology Fellowships, by the Research Corporation through a Cottrell College Science Award (CC5470), and by the National Science Foundation (DUE-0126439).

## Appendix

### Wave Packet Revivals: Theory

Quantum-mechanical wave packet revivals have recently received considerable theoretical and experimental attention [29]. Most often the theoretical research has focused on initially localized wave packets in the infinite square well (ISW) because of its well-known time scales: the classical time scale and the revival time scale. We examine the time dependence of such an initially localized state by choosing a standard Gaussian wave packet of the form

$$\psi_{(G)}(x, 0) = \frac{1}{\sqrt{b\sqrt{\pi}}} e^{-(x-x_0)^2/2b^2} e^{ip_0(x-x_0)/\hbar} \quad (\text{A1})$$

where by direct calculation:  $\langle x \rangle_{t=0} = x_0$ ,  $\langle p \rangle_{t=0} = p_0$ , and  $\Delta x_{t=0} = \Delta x_0 = b/\sqrt{2}$ . The wave packet can be constructed from a sum of energy eigenstates as:

$$\psi(x, t) = \sum_{n=1}^{\infty} c_n u_n(x) e^{-iE_n t/\hbar} \quad (\text{A2})$$

where the expansion coefficients satisfy  $\sum_n |c_n|^2 = 1$ . The expansion coefficients are determined by an 'overlap' integral of the individual energy eigenstates with the initial Gaussian wave packet.

We find that over time the wave packet spreads in a characteristic way defined by the spreading time,  $t_0 = mb^2/\hbar$ . For longer times, we have the so-called classical period,  $T_{cl}$ , and the revival time,  $T_{rev}$ , which are related to each other by:

$$T_{rev} = 2\pi\hbar/|E''(n_0)|/2 = 2\pi\hbar/E_0 = 4mL^2/\pi\hbar = 2n_0 T_{cl}. \quad (\text{A3})$$

The classical periodicity for the ISW agrees with the classically-expected result ( $T_{cl} = 2L/v$  which corresponds to the time it takes a classical particle to traverse the well twice). The so-called initial 'bounce' at  $T_{cl}/4$  with one wall is of interest because of the similarities and differences between the classical and quantum-mechanical cases [21, 22]. Also of interest in the context of the ISW, but on a considerably longer time scale than the classical time scale, are the well-known exact, mirror, and fractional revivals [26-28] in which the wave packet or copies (sometimes called 'mini-packets' or 'clones') of the original wave packet reform long after the original packet 'collapses.' The analysis of these problems often requires specialized visualization techniques because of the long times involved.

## References

- [1] E. Cataloglu and R. Robinett, "Testing the Development of Student Conceptual and Visualization Understanding in Quantum Mechanics through the Undergraduate Career," *American Journal of Physics*, **70**, 238-251 (2002).
- [2] M. Belloni and W. Christian, "Physlets for Quantum Mechanics," *Computing in Science and Engineering* **5**, 90 (2003). The Physlet-based quantum mechanics exercises from this paper are available at: [http://webphysics.davidson.edu/cise\\_qm](http://webphysics.davidson.edu/cise_qm).

- [3] M. Belloni, L. Cain, and W. Christian, "Using Physlets and Open Source Physics to Make Quantum Mechanics Visual and Interactive," *Transformations: Liberal Arts in the Digital Age* Volume 2, No. 1 (2004).
- [4] A. Szabo and N. Hastings, Using IT in the undergraduate classroom: should we replace the blackboard with PowerPoint?, *Computers and Education* **35**, 175-187, 2000.
- [5] R. Beichner, "The Impact of Video Motion Analysis on Kinematics Graph Interpretation Skills," *American Journal of Physics*, **64**, 1272-1277 (1997).
- [6] W. Christian and M. Belloni, *Physlets: Teaching Physics with Interactive Curricular Material*, Prentice Hall, Upper Saddle River, NJ, 2001. See also: <http://webphysics.davidson.edu/applets/applets.html>.
- [7] M. H. Dancy, W. Christian, and M. Belloni, "Teaching with Physlets<sup>®</sup>: Examples From Optics," *The Physics Teacher*, **40**, 40-45 (2002).
- [8] A. J. Cox, M. Belloni, W. Christian, and M. H. Dancy, "Teaching Thermodynamics with Physlets<sup>®</sup> in Introductory Physics," *Physics Education*, **38**, 433-440 (2003).
- [9] W. Christian and M. Belloni, *Physlet<sup>®</sup> Physics: Interactive Illustrations, Explorations, and Problems for Introductory Physics*, Prentice Hall, Upper Saddle River, NJ, 2004.
- [10] M. Belloni, W. Christian, and Melissa Dancy, "Teaching Special Relativity Using Physlets," *The Physics Teacher*, **42**, 284-290 (2004).
- [11] D. N. Blauch, "Java Classes for Managing Chemical Information and Solving Generalized Equilibrium Problems," *Journal of Chemical Information and Computer Science*, **42**, 143-146 (2002).
- [12] G. Novak, E. Patterson, A. Gavrin, and W. Christian, *Just-in-Time Teaching: Blending Active Learning with Web Technology*, Prentice Hall, Upper Saddle River, NJ, 1999.
- [13] E. Mazur, *Peer Instruction: A Users Manual*, Prentice Hall, Upper Saddle River, NJ, 1997.
- [14] T. O'Kuma, D. Maloney, and C. Hieggelke, *Ranking Task Exercises in Physics*, Prentice Hall, Upper Saddle River, NJ, 2000.
- [15] L. McDermott and P. Schaffer and the Physics Education Group, *Tutorials in Introductory Physics*, Prentice Hall, Upper Saddle River, NJ, 2001.
- [16] R. Hake, "Interactive-engagement vs. Traditional Methods: A Six-thousand-student Survey of Mechanics Test Data for Introductory Physics," *American Journal of Physics*, **66**, 64 (1998).
- [17] D. Sokoloff and R. Thornton, "Using Interactive Lecture Demonstrations to Create an Active Learning Environment," *The Physics Teacher*, **35**, 340 (1997).
- [18] B. Thacker, "Comparing Problem Solving Performance of Physics Students in Inquiry-based and Traditional Introductory Courses," *American Journal of Physics*, **62**, 627 (1994).
- [19] Francisco Esquembre, Ernesto Martín, Wolfgang Christian y Mario Belloni, *Fislets: Enseñanza de la Física con Material Interactivo*, Prentice-Hall, España, 2004.

- [20] The Open Source Physics code library, documentation, and curricular material can be downloaded from the Web site: <http://www.opensourcephysics.org/default.html>.
- [21] M. Andrews, "Wave packets bouncing off walls," *Am. J. Phys.*, **66**, 252-254 (1998).
- [22] M. A. Doncheski and R. W. Robinett, "Anatomy of a quantum 'bounce'," *European Journal of Physics*, **20**, 29-37 (1999). M. Belloni, M. A. Doncheski and R. W. Robinett, "Exact Results for Gaussian 'Bouncing' Wave Packets," accepted to *Physica Scripta* (2004).
- [23] R. W. Robinett, "Visualizing the Collapse and Revival of Wave Packets in the Infinite Square Well Using Expectation Values," *Am. J. Phys.*, **68**, 410-42 (2000).
- [24] F. Grossmann, J.-M. Rost, and W. P. Schleich, "Spacetime Structures in Simple Quantum Systems," *J. Phys.*, **A30**, L277-L283 (1997); W. Loinaz and T. J. Newman, "Quantum Revivals and Carpets in Some Exactly Solvable Systems," *J. Phys. A: Math. Gen.*, **32**, 8889-8895 (1999).
- [25] M. Belloni, M. A. Doncheski and R. W. Robinett, "Wigner Quasi-probability Distribution for the Infinite Square Well: Energy Eigenstates and Time-dependent Wave Packets," *American Journal of Physics*, **74**, 1183-1192 (2004).
- [26] I. Sh. Averbukh and N. F. Perelman, "Fractional Revivals: Universality in the Long-term Evolution of Quantum Wave Packets Beyond the Correspondence Principle Dynamics," *Phys. Lett.*, **A139**, 449-453 (1989); "Fractional Revivals of Wave Packets," *Acta Phys. Pol.*, **78**, 33-40 (1990).
- [27] R. Bluhm, V. A. Kostelecky, and J. Porter, "The Evolution and Revival Structure of Localized Quantum Wave Packets," *Am. J. Phys.*, **64**, 944-953 (1996).
- [28] D. L. Aronstein and C. R. Stroud, Jr., "Fractional Wave-function Revivals in the Infinite Square Well," *Phys. Rev.*, **A55**, 4526-4537 (1997).
- [29] For an extensive review of this topic see: R. W. Robinett, "Quantum Wave Packet Revivals," *Physics Reports*, **392** 1-119 (2004).