

7-2019

AP Physics: Modeling with Computer Simulations

Maxwell Vincent Fazio

Korea International School, maxwell.fazio@kis.or.kr

Follow this and additional works at: https://digitalcommons.trinity.edu/educ_understandings

Repository Citation

Fazio, Maxwell Vincent, "AP Physics: Modeling with Computer Simulations" (2019). *Understanding by Design: Complete Collection*. 450.

https://digitalcommons.trinity.edu/educ_understandings/450

This Instructional Material is brought to you for free and open access by the Understanding by Design at Digital Commons @ Trinity. For more information about this unie, please contact the author(s): maxwell.fazio@kis.or.kr. For information about the series, including permissions, please contact the administrator: jcostanz@trinity.edu.

AP Physics: Modeling with Computer Simulations - Maxwell Fazio

Stage 1 - Desired Results

AP Physics 1 Science Practices Addressed:

1.1 The student can create representations and models of natural or man-made phenomena and systems in the domain.

1.2 The student can describe representations and models of natural or man-made phenomena and systems in the domain.

1.3 The student can refine representations and models of natural or man-made phenomena and systems in the domain.

1.4 The student can use representations and models to analyze situations or solve problems qualitatively and quantitatively.

1.5 The student can re-express key elements of natural phenomena across multiple representations in the domain.

2.1 The student can justify the selection of a mathematical routine to solve problems.

2.2 The student can apply mathematical routines to quantities that describe natural phenomena.

Transfer

Students will independently use their learning to...

Create a computer simulation of a physical system using VPython and utilize their program to conduct an investigation regarding a scientific question of their choice.

As a result of this process students will learn to:

- Translate system models into different formats (i.e. translate a physical/mathematical model into a computer program).
- Independently troubleshoot difficult problems through use of online forums and databases.
- Interpret and evaluate the work of others (i.e. peer/sample computer code) to learn methods and approaches of inquiry.
- Evaluate the feasibility and potential of computational approaches to solving new problems.

Meaning

Understandings

Students will understand that...

- Computer simulations allow scientists to model systems that are too complex to model analytically (e.g. systems with too many interacting objects).
- Computer simulations allow scientists to conduct controlled experiments in a manner that makes it easy to change the parameters (initial conditions) and quickly see how they affect experimental outcomes.
- Computer simulations can be used to make predictions about the behavior of real-world systems.
- Computer simulations can predict the motion of systems containing objects undergoing non-constant forces.

Essential Questions

- Why do scientists and engineers conduct computer simulations?
- How can we predict the motion of physical systems undergoing non-constant forces?
- How do scientists and engineers use computer simulations to model the behavior of systems that include lots of objects?

3.1 The student can pose scientific questions.

3.2 The student can refine scientific questions.

3.3 The student can evaluate scientific questions.

4.1 The student can justify the selection of the kind of data needed to answer a particular scientific question.

4.2 The student can design a plan for collecting data to answer a particular scientific question.

4.3 The student can collect data to answer a particular scientific question.

5.1 The student can analyze data to identify patterns or relationships.

5.3 The student can evaluate the evidence provided by data sets in relation to a particular scientific question.

6.1 The student can justify claims with evidence.

6.4 The student can make claims and predictions about natural phenomena based on scientific theories and models.

7.1 The student can connect phenomena and models across spatial and temporal scales.

Acquisition

Knowledge

Students will know...

- Iterative loops work by steadily increasing the time in steps and updating all relevant variables over each time step by following mathematical constraints of the system.
- The behavior of objects undergoing non-constant forces can be approximated by assuming forces are constant over very short time intervals, updating the force after each interval, and repeating this process over many intervals.
- Indentations in python are used to indicate nested loops.
- Variables can be defined by using “=”
- Variables can be changed by using “+=”
- The how “while”, “for”, and “if” commands can be employed to run loops, print variables at desired times, and terminate loops at desired times.
- Relevant VPython Syntax for:
 - Creating an object (and specifying its elements)
 - Creating vectors
 - Adding/subtracting, vectors.
 - Multiplying a vector by a scalar
 - Modifying the components of a vector.
 - Finding the square of the magnitude of a vector.
 - Computing a unit vector
 - Using # to add comments
 - Printing values of variables

Skills

Students will be able to...

- Create and investigate a scientific question that requires a computer simulation to investigate.
- Write, modify, and debug iterative code to model physical systems including objects undergoing non-constant vector-sum of forces and multiple interacting objects.
- Write, modify, and debug code to display real-time visual simulations of physical systems and print variable values at desired instants.
- Modify existing code to conduct controlled experimentation (e.g. methodically alter the initial conditions to examine changes in experimental outcomes).
- Navigate help resources (dictionaries, videos, and forums) to independently troubleshoot.
- Create and modify object elements and vector components.
- Perform vector and scalar operations, including modifying individual components of a vector.
- Create iterative loops that terminate and/or print values of variables at desired times.
- Annotate code line by line in such that a peer can understand its function.

Stage 2 - Evidence

Evaluative Criteria (for rubric)

[See Rubric \(at the end of assignment sheet\)](#)

Performance Task(s)

Students will demonstrate meaning-making and transfer by...

Creating an original simulation using VPython Glowscript to simulate a system of their choice. The students must also utilize their simulation to conduct a controlled investigation of a scientific question they have devised.

Other Evidence (e.g., formative)

- Daily monitoring (students submit their code via email for progress monitoring throughout the unit)
- Exit Tickets and Checking are given throughout as described in the learning plan.
- Written feedback on lab work as described in the learning plan.

Stage 3 - Learning Plan

Pre-Assessment

How will you check students' prior knowledge, skill levels, and potential misconceptions?

- Iterative Modeling Activity: Given on first day of the unit, this designed to see if students have the intuition (or background knowledge) to solve a problem using iterations. See "Lesson 1." The instructor should walk around and listen to student conversations to gather anecdotal evidence regarding each students contributions.
- Exit Ticket: Students submit exit ticket via LMS answering the question: What experience do you have with computer programming?

Learning Activities: Note each lesson is designed to take a full 80 minute block. If you have a different schedule, you may need to modify this. Some lessons may take even longer than a full block.

Before looking through the unit, I encourage you to take a look at [this video introduction to Python Glowscript](#).

Lesson 1: An Introduction to Iterative Modeling

Lesson Goal: Students gain an understanding for how iterative computing works and how it can be employed to solve problems. This activity also functions as an informal pre-assessment.

Grouping: Students should be placed in groups of three or four and allowed to work collaboratively using dry/erase boards. Students should be given scientific (but not graphing) calculators.

Iterative Modeling Activity: To probe student's intuitions/backgrounds regarding iterations, they will be tasked with answering the following question: "A 1 kg mass is connected to a spring of spring constant 0.5 N/m. If the spring is displaced .25 m from equilibrium, how long does it take to reach its equilibrium position."

- Note that this is a question that students will be unable to answer with AP Physics 1 approaches because the spring force is non-constant as the stretch distance changes. As students brainstorm ideas and try some approaches, choose appropriate times to give the following hints/tips:
 - Do not try to calculate an exact answer instead, try to *approximate the answer*.
 - The force is non-constant. What if you break the interval up into smaller intervals and pretend the force is constant over each one?
- **Board Meeting:** Gather students in a circle. Student groups present their dry erase boards and explain the approach their groups took and (if they were able to determine an answer) the answer they found. All groups must explain their approaches, even if they struggled a lot with this process.
- Recap: With students, walk through this problem showing how it can be solved iteratively. [Here is an explanation of one way to solve it](#). Additionally, discuss what would happen if the iteration distance were made bigger/smaller. How would this affect the accuracy of the approximation? Discuss with students how iterative computing allows us to solve problems that can not be solved simply by hand. Lead this into a discussion about how this process can be automated through computer coding and doesn't need to be done by hand.
- *Note if students really struggle with this, you may want to actually give them printouts of the sample solution linked above, then ask them to annotate what's happening in each step. It may also be valuable to project it up onto the screen and walk students through each iteration step by step.
- **Early Computers:** Show clip of early computers ([Human Computers](#)--Atomic Heritage), discuss the role many women played in major engineering projects (including the moon landing). If you have the time for it, you could even watch parts or all of the movie "Hidden Figures."
- Show this clip about why scientists conduct ([Real-World Computer Simulations](#)--NASAeClips). To discuss the purpose of computer simulations to lead to a discussion about how they allow us to explore mathematical models that are too complicated to solve by hand.
- Give Exit Ticket: What experience do you have with computer programming?

Progress Monitoring (e.g., formative data)

Monitor student groups and listen for contributions from each student.

Take notes regarding the approaches different student groups took for this activity.

Exit Ticket

Lesson 2: Setting up a Glowscript and Rendering Objects

Lesson Goal: Students learn how to sign in to Glowscript, create a program, render 3D objects and edit the the object elements).

Getting Set Up

- Instruct students to navigate to <https://www.glowscript.org/> and create accounts.
- Allow students some time to look through and play with some of the sample programs.
- Using the Projector show students how to navigate to support resources by clicking the “HELP” link. It should take them [here](#).
- On the left there is a drop down menu that says “choose a 3D object.” Students should select some different objects and read how to render some of them. You can also point out this video ([VPython Instructional Videos 1. 3D Objects](#)) which some of them may find helpful.
- At this point it is also a good idea to point out all of the other helpful resources linked on this page and tell students that they should get used to keeping this page open as they will use it regularly throughout the unit.

Task: Creating a Stationary Model of an Object of Your Choice

Objective: Through combining shapes and manipulating their orientation, color, and position create a stationary model of a complex object of your choice.

- Students should be asked to create a program and title it “StationaryModel.” They will use this program to create their stationary model.
- At the end of class, they should share their models with their groups or even with the whole class.
- *Note: Depending on the culture of your class, you could conduct this lesson as a contest and ask students to vote on which models they think are: most accurate, most unique, funniest, etc.
- At the end of class students should share their programs with you for a formative checkup. Students can share their programs by following the steps:
 - Clicking the “edit” link under your program.
 - Once in the editor clicking the “share or export this program” link.
 - Emailing the link.

Lesson 3: Animating Objects:

Lesson Goal: Students learn to animate objects by writing iterative loops.

Prior to the beginning of this lesson, students should watch both of the following videos:

1. [VPython Instructional Videos 3. Beginning Loops](#)
2. [VPython Instructional Videos 4. Loops and Animation](#)

Check-In

- Show students this [constant velocity motion code](#). Give them printed out versions of the code and ask them to annotate the code by hand, specifying what happens on each line. Students should work individually. (Here is a link to an [annotated version of the same code](#).)
- Tell students to put away their pencils and then discuss with a partner. Collect their individual written work, then walk through it with the class and ensure that everyone can follow.

Task: Animating Objects: Students annotate prewritten code showing a ball bouncing back and forth between two planes and a ball moving with constant acceleration. Then students write their own code for a ball bouncing up and down on the floor, then a ball bouncing back and forth in a room restricted to the xy plane. For the more advanced students, if they have time, they can attempt to animate a ball bouncing around in a 3-D box with motion that is not restricted to a plane. *Here are links to completed versions of the student tasks for reference: [Part 1](#), [Part 2](#), [Part 3.0](#), [Part 3.1](#), (Sample code for 3.2 is not completed).

Students submit their programs for their models via email.

Students submit annotation check-in at the beginning of class.

Students submit their programs from the Animating Objects Task via email. Feedback can be provided as email responses

Lesson 4: Using a Simulation to Solve Problems – Projectile Motion

Lesson Goal: Students learn to modify and refine an existing program to solve problems. Students learn to terminate programs at desired times and print values of desired variables.

Grouping: Individual or pairs.

Task: Using a Simulation to Solve Problems: Students are given working code for a projectile and must to modify it to solve a series of problems that steadily increase in complexity.

Debrief/Check-out Last 20 minutes of class:

- Place students in groups of 3-4. Prompt students to discuss their approaches for these problems.
- Many students will struggle significantly with the last problem. Ask student groups to collaborate in troubleshooting the final problem of this task. At the very end, each group should summarize what they think is the best way to approach it for the rest of the class via debrief.

Exit Ticket: At what point in today’s lesson did you find yourself struggling? Are there any particular aspects of coding so far that you find particularly confusing? What are they? → Depending on student responses, these difficulties can be addressed at the beginning of the next class period.

Lesson 5: The Simple Harmonic Oscillator – A Controlled Investigation Through Modifying Code

Lesson Goal: Students learn to modify and refine an existing program to conduct a controlled experimental investigation.

Grouping: Pairs

Motivating The Lab: Remind students about the iterative activity that we did on the first day of the unit. Walk the students through the reasons why it is a system that is not easy to solve analytically (non-constant force).

LAB: The Simple Harmonic Oscillator with Friction: This lesson is designed to be conducted as a laboratory investigation. Students are given a set of objectives, but are responsible for developing their own methods, collecting data, creating graphs, and making claims about using their data as evidence. I will have students include this in their lab notebooks. I expect them to include the following:

- a restating of the objectives
- clear, repeatable methodologies for each objective (this includes printouts of their code)
- data tables and graphs (printed from a computer and likely generated using Logger Pro)
- A logically argued written claim relying on their data as evidence

*Students are expected to work on their analysis at home after gathering data and working on their code with their partner in class.

Board-Meeting Debrief: After completing their lab work student groups share out their methods and results for objective 2. I suggest following the “board-meeting style”:

- Student pairs summarize their methods and sketch their graphs on the big-size table dry erase boards.
- Student pairs reveal their boards simultaneously we go around the room and each group explains their work (speaking for no longer than 1 minute).

*Based on listening to the work of others, students have an opportunity to revisit and revise their work if they notice anything problematic in their own approaches. Their final work for this will be due the following day.

Gather evidence informally regarding by listening to students during the debrief discussion.

Students can be asked to record their work on paper and submit it at the end of class, but this hampers collaboration, so I encourage them just to work on dry erase boards. (low-stakes)

Exit Ticket

Students submit written-up work for this lab assignment. Written feedback should be given on this work and returned to the student. If general rubrics for basic science practices are utilized on a regular basis, they can be applied to assessing this assignment if you choose.

Lesson 6: Gravitational Interactions – A Deeper Look at Vector Operations Through Annotating Existing Code

Lesson Goal: Students learn how to carry out vector operations.

Grouping: Individual, but students can engage in conversation with a partner throughout. (Motivating question can be discussed in groups.)

Motivating the Task: Pose the problem: Two massive objects are given initial velocities in close proximity to each other. How can we predict their motion? Students brainstorm why this problem is difficult to solve analytically. Additionally: bring up what happens if a *THIRD* body is added to the system. How does this affect the complexity of the system?

Task: Orbiting Bodies: A Deeper Look at Vector Operations:

- Students are given code for a simulation that models two stars orbiting each other and are asked to annotate the code. There is some syntax that they haven't seen before, so this will take some time. Some of the vector operations are also a bit tricky, so this will require significant effort. Students will learn how to take the magnitude and square the magnitude of a vector. They will also learn the syntax for computing unit vectors.
- Students should compare their annotations with partners.
- For the second portion of the assignment, students are asked to add a *third* body into the system. Although this sounds simple, adding a third body adds significant complexity to the code, so this will likely take the rest of class—students may even need to take this home to continue troubleshooting.
- [Here is a link to my program with the the third body added in for reference.](#)
- You may find it necessary to review the code with the third body on the projector and walk through the vector operations with students.

*Note that this simulation is a modified version of one of the example programs on the glow script site.

Lesson 7: Freefall with Air Drag – A Controlled Investigation Through Creating Your Own Program

Lesson Goal: Students create a simulation from scratch and use it to conduct a controlled investigation regarding objects falling under the influence of air drag.

Grouping: Pairs

Motivating The Lab:

- Show students the equation for drag force (see this [real-world-physics explanation](#)).
- Ask students to discuss with their partner why the behavior of an object falling under the influence of drag force is difficult to predict analytically.
- This should lead to a discussion about the fact that the force is non-constant.

LAB: Freefall with Air Drag

- Very little guidance is given for this lab activity. Students are given resources explaining the mathematical model for air drag force. Students are tasked with creating a simulation and conducting a controlled investigation to answer a scientific question regarding the relationship between variables of their choice.
- Expectations for the lab write-up are the same as for the lab given in lesson 5.

Board-Meeting Debrief: Follow the board-meeting protocol described in Lesson 5

Students submit their code via email at the end of class for written feedback.

As with the lab work in Lesson 5, students can submit written-up work for this lab assignment as well. Constructive written feedback should be given. If general rubrics for basic science practices are utilized on a regular basis, they can be applied to assessing this assignment if you choose.

Lesson 8 (Optional): Predictive Modeling –Falling With Air Drag

Lesson Goal: Students use their air drag simulations from the previous lesson to make predictions about real-world objects.

Grouping: Students should work with the same partner they had for the previous lesson.

- This lesson should only be completed if there is sufficient time, but I imagine a lot of students would find it pretty empowering.
- Give students a real-world object like a ping-pong ball, coffee filter, paper cone, etc... It needs to be something that can fall without rotating and experiences significant air drag.
- Allow students to measure the mass and dimensions of the object.
- Students then use their simulations to predict the time it will take for the object to fall a set distance.
- Drop the object and time the fall. Compare the experimental results with the predictions from their simulation.
- Debrief with students focusing on comparing whether the experimental or predict time was greater. Encourage students to closely examine their simulations to uncover the source of this discrepancy.

Board-Meeting Debrief: Follow the board-meeting protocol described in Lesson 5