Flipping the Physics 4A Classroom
Programming in VPython
A Sabbatical Leave Project for Fall 2015-Spring 2016

Phillip Wolf
Professor of Physics
Mt. San Antonio College
September 1, 2016

Abstract

A "flipped" classroom is one in which the information delivery parts of the class (traditional "lecture") happen outside of the classroom, while the problem-solving parts of the class (traditional "homework") happens in class. There are several reasons to expect that students might have an improved experience in a course taught in a "flipped" format.

- Students prefer to control the pace at which they take in new information
- Students who typically "phase" out for a minute or two during a lecture have a way to recover what they have missed.
- Students prefer to do problem-solving in groups, where they can discuss and try different approaches and get help when they get stuck.
- Watching students "do homework" gives the instructor a richer perspective on what concepts students have difficulty with and allows for "just in time" teaching to address those issues.
- When students come to class prepared, more time and focus can be spent in class on answering student questions, having class discussions, and addressing more complex and interesting questions.

During my sabbatical leave I developed the online materials to "flip" my first semester Engineering Physics course, Physics 4A, and to figure out the mechanics of teaching in this format. A subpart of this sabbatical was learning VPython ("Visual Python"), a programming language developed by Physics instructors to program simulations of a variety of physical situations. I wrote dozens of programs to illustrate a variety of concepts. Many of these simulations made their way into the pre-class online "lecture" materials.

A third, small part of this sabbatical involved my studying materials at a level a step higher than what I teach. I took an online Electrical Engineering course through MIT and worked a bunch of problems from a classic Special Relativity textbook.

How this sabbatical serves the college

While the arguments for teaching in a flipped format are very compelling and flipped instruction is a growing educational trend, few instructors at Mt. SAC (or anywhere) teach in this format. It is worthwhile for our students to experience learning in this fashion. And it is useful to the college to have someone who has figured out how to make this format work, useful for the College to have a sense of what it takes in terms of time and resources to switch a course over to this format, and useful to have someone who can teach others and serve as a resource on how to make this work.

I will be sharing with my Department the materials I have developed for Physics 4A. We can develop additional materials for this and other courses, with my experiences serving as a foundation from which to build.

I will be running a Flex Day workshop on flipped instruction before the Spring term to introduce other interested instructors to this method of teaching.

The simulations I have written have been incorporated into my Engineering Physics course. Now that I know how to program in VPython I can write simulations to illustrate concepts in other Physics and Engineering courses.

Table of Contents

TABS

Sabbatical Proposal for Phillip Wolf, Department of Physics and Engineering, for 2015-2016

I have three goals for my sabbatical:
1) Develop a set of online videos/lectures/screencasts*/quizzes/assessment and accountability measures to supplement/reinforce present lecture topics with the ultimate goal of "flipping" my classroom, and learn to use and implement software systems that would make this all work.
2) Learn to program in V-Python (a programming language used extensively in computational modeling of physical systems) and to program an Arduino (a microprocessor used to collect data from electronic sensors and to control basic instruments) to support classroom lectures, lab activities, and student research projects.
3) Solve Physics problems that are a step beyond what we do in our classes with the goal of enriching my ability to present a broader perspective to my students of "what comes next."

PART 1: Flipping my classroom.

Background:

I have been teaching for thirty years (twenty of them at Mt. SAC) and was a homeschooling parent for ten years. This combination of classroom teaching and homeschooling has developed in me a sense that bringing 30 students together to deliver information to them, one time through and all at the same pace, is neither particular efficient nor developmentally appropriate. It is somewhat akin to having a standard that "All children shall be walking at an age of 12 months". Virtually all children eventually learn to walk, but some of them need a little more time. Likewise, although I think most of our physics and physical science students can master the concepts and material that we teach, many of them (especially my nontraditional and disadvantaged students) can't master it at the pace at or format in which we deliver it. What they seem to need, as indicated by how they study, are videos that they can watch at their own speed and that will allow them to catch all of the important details of a lecture, re-watching the parts where they run into trouble. In addition, they also want frequent checkpoints along the way to know whether or not they have understood the important points from the lesson.

There is simply too little time in class to give all of the students this kind of individualized experience. And my experience (and that of my colleagues) is that telling students to "go home and review it on your own" or "come in to office hours if you have questions" is not effective or useful to most of our students.

I have given certain lectures over 40 times during my tenure at Mt. SAC. Even if my lectures are brilliant, the reality is that students mentally check in and out during lectures or are too busy writing to catch every idea or detail.

My observation (certainly not unique) is that students enjoy working on problems together in class. They learn best when they can bounce ideas off of one another. It is my developing sense that the best reason to bring students together at the same place and time is to take advantage of these unique opportunities to interact with the instructor and with one another.

---

* From Wikipedia: "A **screencast** is a digital recording of computer screen output, also known as a *video screen capture*, often containing audio narration. The term *screencast* compares with the related term *screenshot*; whereas screenshot generates a single picture of a computer screen, a screencast is essentially a movie of the changes over time that a user sees on a computer screen, enhanced with audio narration." http://en.wikipedia.org/wiki/Screencast

What seems to me to be a better approach to teaching would be a "flipped" classroom. In a flipped classroom, the introductory lecture material happens outside of class. The students watch a video or do a focused reading and some assessment exercises before coming to class. This "pre-class" material is a significant part of their grade. What happens *during* class is smaller focused lectures and guided group problem solving aimed at clarifying the work that students did outside of class, in addition to the lab activities we already do now. The logic here is that the best reason for bringing students together in one room at the same time is not to talk *at* them, but to take advantage of the synergies that can occur when they are working together. Ideally, when they leave class at the end of the day they have their problem-solving "homework" done. Hence the term "flipped" classroom—the lecture happens outside of the class, and the problem-solving happens in class. The key to flipping a classroom is to have an extensive array of "pre-lecture" materials prepared (usually a videos, online activities, or screencasts) and assessments and accountability tools to ensure that students do (and get credit for) the doing the "homework" before coming to class.

My proposal is to develop the necessary materials and learn the software tools that would make possible teaching in a flipped environment. I have some experience writing this kind of material already in that I have co-written three editions a textbook (*Problem Solving for Conceptual Physics*, with Paul Hewitt). I wrote most of the chapter introductions, each of which is, in essence, a mini-lecture introducing a topic in Physics (Newton's Laws of motion, Energy, Special Relativity, etc.) with worked example problems.

In addition, the Physics and Engineering Department owns a variety of video resources already, but the actual useful bits are scattered among many hundreds of hours of material. Part of my sabbatical project would be to extract out from those hundred of hours the useful sections, to place them in a context that conveys the ideas they contain effectively and efficiently, and then make them available to our students to support their understanding of key concepts.

<u>My Proposal</u> (*Project*):

1. Develop and assemble and identify a series of screencasts/YouTube /Khan Academy-type videos that address many of the topics that we cover in lecture, focus primarily but not exclusively on topics covered in our Physics 4A course. These videos are intended as pre-lecture preparation and post-lecture reinforcement for each topic. The ultimate goal is to "flip" my classroom, and provide enrichment tools that my colleagues who are not teaching in a flipped classroom can still use. My project will include:
   a. Developing original material. (I have already bought a tablet and software that will allow me to make and caption screencasts. It will take some time to learn how to use these.)
   b. Editing resources we already own (Mechanical Universe, hundreds of Physics demonstrations/experiments on video, etc.) to extract the useful parts. Some of these will be investigated for efficient classroom and outside-of classroom use.
   c. Finding existing YouTube video bits and online simulations. Some of these will be investigated for efficient classroom use.
   d. Using existing websites as resources.
2. Investigate and learn an online system that will manage the accountability of students' watching/reading the videos/activities and their responding to them.
3. Investigate and learn a system that will allow students to chat online about with each other about what they are watching and to discuss and answer any questions they might have.
4. Develop a set of questions/problems/activities/quizzes that students will answer/solve/do that go with each resource.

I have had some extensive conversations with John Suchocki, who has developed hundreds of video lectures/screencasts for Chemistry (at www.conceptualacademy.com), and who uses this flipped instruction model in his classes. He has told me that it will take me about 100 hours to produce my first screencast. He has also said that when he is *really* cranking it will take twenty hours for him to produce an 8-minute screencast. So based on the above, developing a sufficient "arsenal" of materials and expertise to flip a class will take a year.

Cal Poly Pomona has extensive resources available to instructors who wish to flip their classrooms, including help on using screencasting and course management software. As an adjunct Lecturer there I have access to (and plan to take full advantage of) those tools.

I have gone through each of our Physics/Physical Science courses and listed many of the topics in each course that we lecture on. (See the attached spreadsheet.) I have identified where the same concept is covered in different courses. I would start with topics in Mechanics that show up in Physics 4A and which also apply in Physics 2AG and other courses.

Benefit to the College: At the end of the sabbatical I will be able to offer all of these materials to the Department and serve as a resource for implementing this stuff in their classes at whatever level they choose to use it, as well as give a Flex Day presentation on how all of this works. I will also be ready to flip my Physics 4A classroom.

Benefit to my professional growth and enrichment: I have been teaching for thirty years and have a decade or more of teaching left in my career. The students I have today approach their learning differently than the students I had when I started teaching at Mt. SAC twenty years ago. I have been feeling for a couple of years that I am due for a paradigm shift in how I approach my teaching. In consultation with my colleagues in my department, I think that developing these materials will lead to better outcomes for students and greater satisfaction for me.

And, I really enjoy developing curriculum! Writing a textbook and seeing it out in the world has been incredibly satisfying. Developing these videos will be the same kind of experience for me.

PART 2: Learn V-Python and how to program an Arduino

Background:

V-Python is a programming language and a powerful tool for computational modeling and for writing simulations of physical systems. Lots of Physics problems can be solved and physical situations modeled and animated in V-Python. I don't know how to program in it at all. If I could program in V-Python I could use it as a tool in class and could write simulations for the screencasts for part 1 of my proposal. In addition, my department has a Computational Modeling course working its way through the EDC. Only two members of my department presently have experience in doing computational modeling in V-Python.

An Arduino (pronounced *Ardweeno*) is a microprocessor/controller capable of controlling and polling a tremendous number of instruments and sensors. Last summer the Physics and Engineering department completed a high-altitude balloon project with about 25 students. We launched a balloon with various sensors and an Arduino to an altitude of about 12 miles. The Arduino was programmed by students to capture all of the data from the flight. What I had suspected before, and what was really driven home for me during the project, is how powerful a well-programmed Arduino can be in designing and carrying out research.

I teach the general education Energy Science course (PHSC 3) in my department. (It may be interesting to note that my expertise in the field of renewable energy, and my ability to develop the Energy Science course, came from my sabbatical in 2002-2003, when I went back to school to get a Masters degree in Renewable Energy and the Environment). My colleague Daniel Anderson has done a phenomenal job sparking interest in our Engineering students in doing renewable energy projects. Almost any long-term research project in renewable energy will involve collecting data over a long period of time and controlling various experimental parameters on the basis of feedback from measurements of the system. A lot of that can be automated and controlled by an Arduino, but I don't know how to do it. So some experimentation and research that we could be doing (and that I could be working with students on) isn't getting done. Presently only one member of my department (Martin Mason) knows how to program an Arduino.

My Proposal (*Self study*):
Harvey Mudd College has put online their computer science course "CS5 For All" (http://www.cs.hmc.edu/csforall/). A significant part of the course is learning to program in V-Python. So, another part of my sabbatical would be working through that course in V-Python. In addition, lessons in programming an Arduino are available online at http://www.ladyada.net/learn/arduino/. I will work through these lessons as well.

Benefit to the College: I would be able to contribute to curriculum development for the new computational modeling class; to write simulations to use in screencasts and in class; to more effectively include computational modeling in my present physics classes; to develop lab activities for the Energy Science course that would include long-term data collection and active control systems (solar tracking, for example); and to more effectively mentor students doing research projects.

Benefit to my professional growth and enrichment:. Tools are an extension of the mind. Having a garage full of tools that you know how to use opens up an entire world of creative possibilities that don't exist without those tools and the knowledge of how to use them. For me, one of the most enjoyable parts of teaching is the ability to create new ways of presenting ideas and materials to my students. Having more programming tools at my disposal would allow me to be a more effective and creative teacher and scientist.

PART 3 (a small part): Solving Physics problems that are harder than what I teach.
Background:
I have been teaching for thirty years. I am good at solving most of the problems that come up in the Physics courses that we teach. I am not so good at solving the next level of problems beyond the realm of what we teach. I think I would be a better physicist if I spent time doing what I tell my students they need to do to become better physicists--solve lots of challenging problems!

There is a set of textbooks (the The *M.I.T. Introductory Physics Series*) that came out in the early 1970's that covers some of the Engineering Physics material at a slightly higher level than our textbook does. In the 1960's a group of professors at UC Berkeley wrote the *Berkeley Physics Course*, a mathematically sophisticated approach to the first two years of university-level Physics. Also in the 1960's, the Nobel-winning physicist Richard Feynmann taught the freshman-sophomore physics courses at Caltech and his lectures were transcribed (*The Feynmann Lectures*). There were challenging problem sets developed to go with his 117 individual lectures.

The last part of my sabbatical would involve reading through some of this vast amount of material and working a lot of Physics problems. Doing this takes time!

Benefit to the College and to my profession growth and enrichment: I would be a better and more confident physicist. I would have a better sense of perspective of "what comes next" when I am working on materials for the classes I teach. I could better encourage students with capability and interest beyond the level of the courses that I teach now. And I would get to reconnect with my field beyond the confines of what I teach.

Academic and other references:

Flipped instruction:

Andrew Westphal, UC Berkeley Physics Department on UCB's Center for Teaching and Learning Blog: http://teaching.berkeley.edu/blog/experiment-flipped-physics

James Madison University article: Flipping the Classroom improves student learning in Physics: http://www.jmu.edu/news/2014/05/30-flipped-classroom-improves-learning.shtml

Vanderbilt University Center for Teaching article about Flipping the Classroom http://cft.vanderbilt.edu/guides-sub-pages/flipping-the-classroom/

Articles on flipped classrooms from the Flipped Learning Network: http://www.flippedlearning.org/site/default.aspx?PageID=1

Links to videos and articles about flipping the Physics classroom: http://www.flippingphysics.com/flipping.html

A high-school Physics teacher's home page about how her flipped classroom works: http://www.kgphysicsclass.com/the-flipped-classroom

V-Python and Arduinos:

Physical Sciences Resource Center—Introductory materials for learning and using V-Python in physics computational modeling: http://www.compadre.org/psrc/items/detail.cfm?ID=5692

Harvey Mudd College's CS5forAll introductory CS course, free and online, which is largely based on programming in VPython: http://www.cs.hmc.edu/csforall/

Arduino Tutorial—Learn Electronics using Arduino!: http://www.ladyada.net/learn/arduino/

Adafruit.com's Arduino Tutorial lessons: https://learn.adafruit.com/category/learn-arduino

Information about older Physics Textbook Series:

The Feynmann Lectures: http://www.feynmanlectures.caltech.edu/
Information on problem sets to accompany the lectures: http://www.feynmanlectures.info/

The Berkeley Physics Series: http://en.wikipedia.org/wiki/Berkeley_Physics_Course
The M.I.T. Introductory Physics Series: http://www.crcpress.com/browse/series/crcmitintphy

Timeline for Phillip Wolf's proposed Sabbatical Proposal

Preliminary comments:

1. I don't know how long it will take to produce a video/screencasts and all of the supporting materials. Based on what I have been told by John Suchocki, who has made hundreds of these, I should expect each screencast to take from 20 hours once I am very good at it (he has been doing this for years) to 100 hours (for my first one). I will use an estimated time of 40 hours per screencast, including development of supporting documents and entering them into the software system, after an initial block of 100 hours to produce my first screencast. This schedule is aspirational rather than fixed. Should it turn out that I can produce screencasts more quickly I will happily produce more of them. It is entirely likely that I will choose to produce simpler screencasts in order to accelerate their development.
2. The contract seems to suggest that sabbaticals are for Fall (late August to December), Spring (late February to mid-June), or Spring and Fall only (August to December and late February to mid-June). Judging by my reading of the contract, these are the only times I should be scheduling for my sabbatical work. I am *not* being difficult or doctrinaire in this at all. I will likely do work through the Winter term as well. The sabbatical application said that I should read the contract on the subject of sabbaticals, and that is what the contract seems to say. So, based on what I know, those are the time periods for which I should be preparing a timeline. I welcome another interpretation of the contract language.

Month	Activities
Late August-September	1. Script first screencast and assessment activities(Vector Addition) 2. Work through Camtasia screencasting software tutorials, with assistance from Staff Development at Cal Poly Pomona. 3. Work through Painter X3 drawing software tutorials 4. Prepare first screencast. 5. Research software platforms for integrating the assessment exercises with the screen casting. (Moodlerooms? Cengage product? Something else?)
October	Research existing online resources for screencasts for 1. Unit Conversions 2. Significant Figures 3. Propagated Error Calculations 4. Algebra-based derivation of kinematics equations 5. Calculus-based derivation of kinematics equations.
November	Start CS5 for All course online Work through the first four tutorials on the Arduino Produce screencasts and supporting materials for 1. Position vs. Time graphs 2. Velocity vs. Time graphs 3. Conceptual foundation of Newtons 2nd Law
1st half of December	Continue CS5 for All course Work through the next four tutorials for the Arduino Screencast Projectile motion

Late February-March	Finish CS5 for All course Work through the next four tutorials on the Arduino Start reading and working through Textbook problems in Kinematics Produce screencasts and supporting materials for 1. Circular orbits and apparent weightlessness 2. Centripetal force equation derivation and applications
April	Continue working through Textbook problems in Newton's Laws Produce screencasts and supporting materials for: 1. Springs 2. Basics of static and kinetic friction 3. Conceptual basis of the Impulse Momemtum theorem 4. Conceptual basis of the Work-Kinetic Energy Theorem
May:	Continue working through textbook problems in Work and Energy Produce screencasts on applications of calculus 1. Gravitational forces 2. Volumes of solids 3. Center of mass of continuous bodies 4. Rocket variable mass problems
First half of June	Work textbook problems in Rotational Motion Produce screencasts and supporting materials for 1. Calculating moments of inertia 2. Torque and Angular Acceleration 3. Simple harmonic motion

The following spreadsheet lists the vast majority of topics that are covered in our Physics and Physical Science courses.

I set them out in a grid, seeing where there might be some overlap in topics are covered in multiple courses and where a single set of screencasts (or modifications of a single set) might possibly be used over a variety of courses.

Although my sabbatical is only about preparing to "flip" our Physics 4A course, I wanted to lay out how this might work on a department level if we were to decide to use screencasts for topics other than those that are included in Physics 4A.

Screencast Topics Page 1	PHYS 4A	PHYS 2AG	PHYS 1	PHYS 2BG	PHSC 9	PHYS 4B	PHYS 4C	PHSC 3
Unit conversions	X	X	X					X
Vector addition	X	X						
Vector dot products	X							
Vector cross products	X							
Significant Figures	X	X						
Propagated error calculations	X							
Deriving Kinematic equations	X	X	X					
Position vs. time graphs	X	X	X					
velocity vs. time graphs	X	X	X					
Calculus derivation of kinematic equations	X							
Conceptual Foundation of Newton's Laws and why we draw free body diagrams	X	X	X		X			
Newton's Third Law	X	X	X		X			
Projectile Motion	X	X	X		X			
Circular orbits and apparent Weightlessness	X	X	X		X			
Centripetal Force equation derivation	X	X	X					
Relative Velocity	X							
Basics of static and kinetic friction	X	X	X					
Definition of work	X	X	X		X			X
Conceptual Work-KE Theorem	X	X	X		X			
Conservation of energy	X	X	X		X			X
Conceptual impulse-momentum theorem	X	X	X		X			
Newton's 3rd law and momentum conservation	X	X	X		X			
Spring Constant	X	X	X					
Elastic Potential Energy	X	X	X					
Gravitational potential energy	X	X	X		X			X
Using calculus to find volumes of solids	X							
Finding center of mass for discrete masses	X	X						
Finding center of mass for continuous distributions	X							
Conservation of momentum and relative velocity problems	X							
Rocket variable mass problems	X							

Screencast Topics Page 2	PHYS 4A	PHYS 2AG	PHYS 1	PHYS 2BG	PHSC 9	PHYS 4B	PHYS 4C	PHSC 3
Radian Measure	X	X						
Rotational Kinematics/Connection with translational quantities	X	X						
Torque and equilibrium problems	X	X						
Using calculus to find moments of inertia	X							
Gravitation as an inverse square Law	X	X	X					
Proof that, for spherical mass distributions, all of the mass can be considered as being at the center	X							
Derivation of an expressions for Gravitational potential energy	X							
Using Calculus to find the gravitational attraction due to continuous bodies	X							
Orbits/Hohmann transfer orbits	X							
Energy calculations for orbits	X							
Periodic Motion--Requirements for SHM	X	X	X					
Deriving the Period for an Oscillating System	X	X						
Small angle approximation for pendulums	X							
Physical Pendulums	X							
damped oscillators	X							
Archimedes' Principle and buoyancy		X					X	
Bernoulli's principle		X					X	
Mechanical Waves		X			X		X	
Types of Mechanical Waves		X			X		X	
Wave reflection/transmission at boundary		X					X	
Wave interference		X			X		X	
Standing waves		X			X		X	
Sound waves		X			X		X	
Sound intensity		X					X	
Superpositions of waves		X			X		X	
Harmonics/Overtones/Musical sound		X			X		X	
Doppler Effect		X			X		X	
Kinetic theory/particle model of gases		X			X	X		X

Screencast Topics Page 3	PHYS 4A	PHYS 2AG	PHYS 1	PHYS 2BG	PHSC 9	PHYS 4B	PHYS 4C	PHSC 3
Evaporation as a cooling process		X			X	X		X
Specific heat capacity		X			X	X		X
Maxwell-Boltzmann Distribution						X		
Molar heat Capacity		X				X		
Avagadro's number--What is a mole		X				X		
Mechanisms for heat transfer		X			X	X		X
Blackbody Radiation				X	X	X	X	X
Greenhouse effect					X	X	X	X
Coulomb's Law				X	X	X		
Electroscopes and polarization			X	X	X	X		X
Electric Fields				X		X		
using Calculus to find electric field due to continuous charge distributions						X		
Conductors and insulators			X	X	X	X		X
Gauss' Law and electric flux				X		X		
Calculus and Gauss' Law						X		
Electric Potential and Electric Pot Energy				X		X		
Calculus and Electric Potential						X		
Water Analogy for DC circuits--V, I, R, P			X	X	X	X		X
Resistance and Resistivity				X		X		
Voltage and current in series circuits			X	X	X	X		
Voltage and current in parallel circuits			X	X	X	X		
Equivalent Resistance			X	X		X		
Kirchoff's voltage and current laws				X		X		
capacitance				X		X		
dielectric constant				X		X		
capacitors in parallel and series				X		X		
Right hand Rule for forces on charges				X		X		
Magnetic force on a current-carrying wire			X	X	X	X		X
How an electric motor works			X	X	X	X		X
Hall Effect				X		X		
Biot Savart Law						X		
Ampere's Law				X		X		

Screencast Topics Page 4	PHYS 4A	PHYS 2AG	PHYS 1	PHYS 2BG	PHSC 9	PHYS 4B	PHYS 4C	PHSC 3
Magnetic fields due to various current distributions						X		
Faraday's Law and induction experiments			X	X	X	X		X
Lenz's Law				X		X		
How a generator works			X	X	X	X		X
How transformers work			X	X	X	X		
Applications of Transformer equation			X	X	X	X		X
Why we use transformers			X	X	X	X		X
Three-phase Power			X	X	X	X		X
Inductances				X		X		
AC circuits--rms voltage and current				X		X		
Reactance				X		X		
Phasor diagrams				X		X		
Impedance				X		X		
RLC Circuit behavior and resonance				X		X		
Electric Fields				X				
Displacement current						X		
Maxwell's equations						X	X	
Electromagnetic waves				X		X	X	
Reflection			X	X			X	
Refraction--index of refraction			X	X			X	
Critical angle/Total internal reflection			X	X			X	
Scattering of light				X	X		X	
Thin Lenses			X	X			X	
Curved Mirrors				X			X	
Drawing Ray Diagrams			X	X			X	
Magnifiers/Optical instruments				X			X	
Huygen's Principle				X			X	
Derivation of Interference equations				X			X	
Derivation of Diffraction equations				X			X	
Thin Film interference				X			X	
Edit Relativity Videos				X			X	
Lorentz transformation and Time Dilation				X			X	

Screencast Topics Page 5	PHYS 4A	PHYS 2AG	PHYS 1	PHYS 2BG	PHSC 7	PHYS 4B	PHYS 4C	PHSC 3
Relativistic Conservation of Energy				X			X	
Relativistic Conservation of Momentum				X			X	
Photoelectric effect				X			X	
Absorption and Emission Spectroscopy				X			X	
Compton Scattering				X			X	
Rutherford Scattering/the Nucleus				X			X	
Wavefunctions/The Schroedinger equation							X	
Particle in a potential well							X	
Potential barriers/tunnelling							X	
Energy bands/conduction theory				X			X	
Semiconductors				X			X	
Describing the nucleus--atomic #, mass				X	X		X	X
Origins of radioactivity				X	X		X	X
Radioactive decay reactions				X	X		X	X
Half-life and nuclear stability				X	X		X	X

Introduction--Where does this sabbatical come from?

I have been teaching for 31 years. I have been feeling for the past couple of years that I am due for one more paradigm shift in my teaching career.

There are lectures that I have given 40 times or more. For the past many years I have wanted to get the lecture done "just right" one time, and then give that "just right" lecture over again the following semesters instead of recreating it from scratch each time.

Each time when I have paused to let students catch up on writing stuff down I have recognized that when they are struggling to copy down information from the board they are not spending time processing it.

Each time I lecture I am aware that students will "phase" out for a minute or so (just as I phase in and out of a lecture or a meeting), and if that "phasing out" happens to occur at a critical point in the lecture, their opportunity to understand the material can be lost forever.

Each time I solve problems on the board and plug in numbers and calculate (or having students calculate) answers, I recognize how boring this can be.

Each time I have heard students venting their frustrations at spending hours stuck on a physics homework problem, I wish that they would do their homework in groups, where they could rely on each other to help get them unstuck.

I recognized that a lecture is a story where concepts are built up one at a time, and that a student who misses a step or struggles with a concept at the particular moment when that concept appears does not have the opportunity to go back and recapture what they missed.

I had heard students talking about how, for their other courses, they would get through these other classes by going on YouTube and finding a video that covered what they didn't understand from class.

And so I imagined that it if I could put together and "record" an excellent lecture and get all of what I wanted into it, students could watch *that* before class. That way, if a student phased out for a moment or got confused by something, they could go back and rewatch that section of the lecture over again.

And I imagined that if I did a good job on what I produced, my colleagues facing the same challenges that I face in making material and ideas and techniques available to students could use them in their own classrooms and that their students would benefit from my work.

And, if this turned out well, I could share how to do this with other faculty at Mt. SAC.

So the puzzle I was contemplating was how to create a learning environment that would address each of these observations and desires. How do I move into teaching in this new paradigm?

A flipped classroom

The short version, or "elevator pitch" for the idea of a "flipped classroom," is this: Students prefer to take information in at their own pace and on their own schedule and they prefer to solve challenging problems in an environment where they can get help when they get stuck. So gathering students together in one place, at one time, just to deliver information to them and then to send them off to do homework problems on their own (which is what a "traditional" lecture class does) is exactly the *opposite* of how most of our students would prefer to function. It would make more sense to let students take in information at their own pace (although still in the context of the class structure following a fixed schedule) and to let them do "homework" in a supportive group environment.

In order for this to work I would need to prepare the "information delivery" part before class so that students would have the lecture materials available to watch at home, and then I would have to figure out a way to make sure that the students actually watched the lectures. This requires creating the lecture videos that students would watch before class, working out how to distribute the videos, and putting together an accountability mechanism to make sure that students actually watched them.

That, in a nutshell, was the main focus of my sabbatical.

Further context:

The course I chose to "flip" is our first semester calculus-based Engineering Physics course, Physics 4A. It has a Math 181 (calculus 2) co-requisite. Students in this course have already taken a semester of Physics 2AG, a trigonometry-based course that in its first ten weeks covers at an introductory level the material that is covered in greater depth in Physics 4A. Physics 4A looks at the more challenging problems that can be done and concepts that can be considered when you have a more sophisticated set of mathematical tools (calculus) and when you already have a fundamental grounding in the basic physics concepts. For many students the pace of "information intake" in Physics 2AG is akin to drinking Physics from a fire hose, and although they have seen the concepts involved they have generally not mastered them.

An interesting feature of our Physics 4A course (all of our courses, actually) is that it is taught in an integrated format in a lecture/lab room. The course is officially scheduled as four lecture hours and three lab hours each week. But because we "own" the students for seven hours a week in a lecture/lab classroom, we have the flexibility to "mix up" lecture, discussion, group problem solving, activities and labs in a way that is more natural to the learning process than "I am going to lecture at you for two hours. Then you are going to spend and hour and a half working on a lab."

My readings about the "flipped" format and conversations with folks who already teach this way suggested that the flipped format works best when you have exactly this flexibility to mix up activities and to present problems and labs in context.

<u>Specific questions I needed to answer:</u>

What will these lectures look like?

- Some people make videos of themselves giving a lecture.
- Some people narrate a PowerPoint and make occasional scribbles on it.
- Some people use their phone to capture a narrated video of them writing stuff down on a piece of paper.
- Some people have done complex 3-D computer animations of what they are narrating.
- Some folks have complex graphics.
- Some folks use "green screens" to insert themselves into other backgrounds (such as placing themselves at the top of a mountain, or in a forest, all from the comfort of their own homes.)
- Some folks have a box in the corner of the screen showing their face as they narrate whatever is happening onscreen.
- Some folks (Khan Academy, for example) present a screencast* from a tablet with a once-through voiceover of whatever is being writing on a tablet, and then put that out into the world without any editing whatsoever.

Suffice it to say, there are many models to choose from.

Related to the above is:

How will I make these videos/screencasts? What technology/platform will I use?

Before the start of my sabbatical I had bought a Wacom Intuos tablet, which is something that graphical designers use. It was the tool of choice for one screencasting guru I know who has produced over a hundred high quality Physics screencasts. I had also purchased with it a copy of Corel X Draw (a graphics software package). The campus has a site license for Adobe Illustrator and other Adobe products. I purchased a *Swivel*, onto which you can mount an iPhone and which (in theory) allows you to wear a necklace so that the iPhone will point toward you and follow you as you walk around and do a lecture. It is like a cameraman that follows you around as you do a lecture.

My initial thought was to assemble the lectures into videos using *Screenflow* on my Mac, and to take advantage of resources available at Cal Poly (where I am a lecturer in the Fall) specifically focused on screencasting.

What is the appropriate split between topics covered in the lectures and topics covered in class?

My initial thought here was to try to cover "everything", but initially focusing particularly on "tedious topics"—ones where I find myself lecturing for half an hour at a time and which involve writing lots of equations on the board—and on lectures that demonstrate a problem solving technique that, in my experience, students have problems with.

---

* A "screencast" is a movie made by capturing whatever is on a computer screen, usually with an accompanying voice-over or narration. The simplest example of a screencast might be a video recording of a narrated PowerPoint presentation. A more complex screencast might involve showing a step-by-step process of how to use a piece of software.

How/where will these videos be hosted?

The idea is that students will be watching these lectures on some platform. The Mt. SAC servers have limits on how large of a file they can manage. The advice I got from the OLSC (Online Learning Support Center) and Learning Resources folks was to host them on YouTube.

How do I make sure that students actually watch the videos before coming to class?

An option suggested by Michelle Newhart in the OLSC was to put together a "Lesson" on Moodlerooms. This would consist of an ordered set of videos, links, and check-in questions and quizzes that would send students through things in order and monitor whether or not students watched the videos and got from them the appropriate information.

It was made very clear to me from conversations with folks who *have* flipped classrooms that watching the before-class material has to be a significant enough part of a student's grade that they will not skip watching it.

How will these videos fit into the overall plan of how a class day goes?

This relates in part to how deeply into the material the pre-class lectures go. But essentially, the plan was to use the time previously spent on "lecture" on "homework" and in-class problem solving, keeping the activities and labs essentially the same. It is really just "flipping" or interchanging the lecture and homework parts of the course.

How will I know if this works?

My expectation is that this approach will have the biggest effect on my struggling students by providing them the structure and support (being able to re-watch stuff they were confused about and having other folks around when they are doing homework, and making doing homework a default activity of the course.) Affectively, students should be less stressed out. More of my D and C students should be doing better and be more engaged.

From expectations to execution

Much of my sabbatical was spent answering those questions, figuring out technology, preparing screencasts, taking some online courses, working with an Arduino, and doing some Physics problems. I do not think it would be useful to attach every movie I made or to print out every program I have written or to include the hundreds of pages of scripts I have generated for screencasts or copies of the problems I have solved, although I will include samples of those and the course I took in the narrative section. My sabbatical was as much "educational" as "project". My intent *here* is to describe many of the paths I followed, the various dead ends and distractions I encountered along the way, and ultimately the choices I made, and why. Were I to start this process again today, knowing what I do now, I would doubtless be more efficient. That is the nature of the learning process.

There were also a few other bits included in my sabbatical proposal:
I would take Harvey Mudd College's "CS for All" course online to learn how to program in Python.

My original plan was to first learn how to program in Python so that I could do cool physics projects with my students, similar to a couple of projects some of my colleagues do with their students, and to learn to write Physics simulations in VPython that could illustrate and animate some difficult physics concepts. (Often in teaching Physics we end up drawing a sketch of some physical situation and asking students to imagine what is going on.) My thinking was that what we are trying to do is a very poor and inefficient way of trying to get the picture in our minds to our students. What we really want is a kind of Vulcan mind-meld. I imagined using that it would be much more efficient if I could just directly show my students the images and animations that are in my head. VPython offers the possibility of doing some of that.

I would learn how to program an Arduino (a microprocessor that can be used to control lights and motors and to collect and record data from a variety sensors).

My plan here was to work through the introductory lessons on how to use an Arduino from a particular website.

I would do some challenging Physics problems, a bit above the level of what we usually do in class.

This was mostly for personal enrichment, and to remind myself what the next level of problems after the Physics 4 sequence look like.

Summary Narrative

What I did—*<u>Programming</u>*

My very first step was to sign up in July 2015 for Harvey Mudd College's "CS5 for All" computer science course through edX, a consortium of colleges and universities that put some of their classes online. The first topic listed in the course description is "Basic Python Programming." Since my main goal was to learn programming in Python (which would put me in a good position to program in VPython), this seemed wonderfully appropriate. As a Harvey Mudd alum I expected it to be a rigorous but satisfying course. The course description said it would take 14 weeks at a pace of 5-7 hours per week, so that seemed like a reasonable amount of time to spend. It would also give me the experience of *taking* an online course, with alternating videos, readings, activities to do and a discussion board, and to see what would work for me and what would not.

I have an innate faith in instructors and course designers. I assume that when they design a course they have a big plan in mind, and that I should approach their course with a faith that I will eventually see how it fits together at the end, even if I don't immediately see why something is relevant. So I did every assignment and every reading, even when they weren't directly related to programming in Python. As I worked through the course I learned a lot about programming in Python. More than once I spent a day and a half trying to solve a single programming problem whose resolution turned out to be two lines of code. Sometimes this took so long because the problem being worked out was a puzzle that had a clever solution that was simple in retrospect, but for which I had to figure out the trick. Sometimes I would scan through the discussion boards trying to see if anyone else had gotten stuck on the same things. Sometimes I would post questions and get answers from one of the Community TA's who reviewed the discussion board postings a couple of times a day. Sometimes the question being asked was framed poorly (so it wasn't clear what was wanted) or just plain wrong (the question said explicitly that we *shouldn't* use a particular method to solve it, when the TA's said that that was the method to use.)

This course ended up taking about three times as much time as advertised♦—an average of 25-30 hrs each week over a period of ten weeks.♣

I ended up spending a lot of time (especially in Spring) working through the details of programming in *VPython*. VPython ("Visual Python") uses the Python language and programming structure, but includes object (arrows, cylinders, boxes, helixes, text, graphing, etc.) that can be animated on a computer screen. Essentially, if you can write a mathematical model for a physical process you can develop a visual, animated representation of it. Learning how to use VPython combines reading through the documentation and example programs with just trying stuff out to see how it works. It was not unusual to spend an entire day (or more) figuring out how to produce a 20-second animation.

One company (Enthought) produces its own version of Python and has a series of online tutorials on using Numpy (a "library" of mathematical and graphing functions that allow for heavy duty mathematical data processing and beautiful publication-quality graphs in a variety of formats.). I spent some days working through these tutorials before abandoning them as being too technical for what I was trying to achieve.

What I did—*<u>Making and posting videos</u>*

In the beginning of the Fall 2015 semester I took the OLSC courses on Moodlerooms♠ and then got my SPOT (Skills and Pedagogy for Online Teaching) certification for online teaching. I am not actually going to be teaching an "online course", but I thought that the tools in the certification would be useful. I got answers to the "How to set up an online discussion in Moodlerooms" question and "How to set up a directed sequence of activities and document that students have done them" question.

I explored a variety of ways of ways to make screencasts. I have a copy of Screenflow, which is screencasting software for the Mac. I watched the *Introduction to Screencasting* course on Lynda.com✢, and more YouTube videos on screencasting than I care to remember. I watched a wide variety of other screencasts, including some very slick videos that used sophisticated special effects and much simpler videos where people mounted their phones three feet above a

---

♦ Interestingly, this was my experience with each of the three online courses that I took. Whatever time estimate was given for the course, the reality was that the official estimate needed to be multiplied by three. I was not slow, but I *was* thorough!

♣ I chose not to take the final exam or do the final project for the course because I had already accumulated enough points in the course that these two things were unnecessary to complete the course requirements, and because spending even more time on the course would have distracted from my goals.

♠ *Moodlerooms* is the campus' Learning Management System. For my purposes, it allows me to post videos, scripts, assignments, and other materials for online distribution to students. Moodlerooms is being dropped entirely by the campus in favor of another system, *Canvas*. The Canvas rollout is to be completed by Summer 2017.

✢ Lynda.com is a professional-level online repository of "How to Do . . ." video lessons on a huge variety of topics. It is especially rich in videos on how to use a variety of computer applications and how to accomplish computer-based tasks. Mt. SAC employees have access to Lynda.com through the *Employees* link on the Portal.

desk and made a video of themselves writing on a piece of paper as they spoke their way through things.
I borrowed, and eventually purchased, an iPad. I looked at a variety of iPad screencasting apps (*OneNote*, *Evernote*, *Explain Everything* and some others) before settling on using *Doceri*.
I spent some weeks figuring out how to use Doceri efficiently and effectively. After two weeks of working on my first presentation using *Doceri* was still awkward. I tried doing the same presentation on PowerPoint but found that awkward as well. Eventually I married the two, taking advantage of what each does well. (A later section of this report describes what technology and formatting decisions I came to and some of the nuts and bolts of producing a screencast video).

During the Fall term I wrote the initial scripts for the early screencasts and tried to marry those with the presentations I was putting together. I met with a member of the Arts Department to find out about drawing and animating things in Adobe Illustrator and Adobe Aftereffects, and spent a week or more working through the Lynda.com course on Adobe Illustrator. I used Illustrator to draw some of the objects that appear in the lecture videos, but ultimately I decided that learning how to use Illustrator properly was going to consume *way* too much time.

Eventually I settled on using Doceri and PowerPoint to create videos, animating things in VPython or PowerPoint, and drawing objects using the drawing tools in Word, PowerPoint, or (rarely) Illustrator or downloading them from "Free Image" sites on the web.

I explored a variety of video hosting alternatives. I ended up using 3Cmediasolutions.com (a video hosting service provide by the Chancellor's office) to host the videos. Students access these links through Moodlerooms. (More on this in the Technology Choices section of this report.)

What I did—*<u>Other Learning Activities</u>*
A third, smaller part of my sabbatical proposal involved doing some physics problems from topics one step beyond what we usually teach in our Engineering Physics courses at Mt. SAC.

I took two additional online courses through edX—the same online course platform through which I took the programming course. One was an *Electrical Engineering* course from MIT, and the other was a *Behavioral* Economics course through the University of Toronto. Details about some self-study on Special Relativity and working with Arduino microprocessors are in a later "Other Learning Activities section of this report.

Time spent:
I starting working on this project in early July 2015. I continued working on it up to the week before the start of school in August 2016. I took perhaps ten weeks entirely off during that time for other non-sabbatical-related activities, and averaged about 30 hours per week for the remaining 46 or so weeks for a total of over 1300 "work-hours", equivalent to 32 weeks of the forty-hour workweek expected in our full-time contract.

CONCLUSIONS/IMPRESSIONS

I won't actually know until the end of a semester or two of teaching whether this flipped classroom approach works. I *can*, however, draw some conclusions about taking online courses, independent learning, and making screencasts!

Online Courses

I love the structure that a well-designed online course brings to learning something I am enthusiastic about. A professional educator, knowledgeable in their field, has already figured out a logical progression of topics and activities and arranged them in a sequential, engaging matter. The opportunity to earn a certificate for completing the course was a good motivation for slogging through some of the more difficult or tedious parts of each course.

The online courses I took through edX also served as examples of different ways of presenting material online.

I also love how taking an online course *not*-for-credit gives one the freedom to decide what to do and what not to do. For one course (the Behavioral Economics course) I decided not to take the final, finish the last lesson or earn the certificate. Instead I went through 85% of the course and bought and read the book the course was based upon. Once I had gotten from the course what I wanted to know, the formality of completing the course was no longer interesting to me.

Likewise, I love the opportunity to try (and abandon) various topics without consequences. I did about 85% of the *Adobe Illustrator* course on Lynda.com before deciding that I did not want to spend the time required to become facile at using that software.

Programming

It is possible to spend a tremendous amount of time trying to write a program that will do what you want it to. I spent over a week on one 20-second simulation, learning how to get the computer to draw graphs that generated themselves as the simulation evolved. I spent days going back and forth between the documentation and coding, trying to learn how to draw a simple ramp in 3D.

Programming can be fun! Part of the draw of programming is a faith that eventually you *can* get the computer to produce what you want. The answer you seek might be just one elusive line of code away.

Indeed, I was eventually able to animate just about any simulation I put my mind to coding.* And I ended up spending an inordinate amount of time doing just that. Part of the pleasure of a sabbatical is flexibility to pursue interesting challenges and puzzles, and for me the simulations I was trying to animate were both interesting and challenging. In retrospect, relative to the amount of time needed for screencasting, I probably spent "too much" time on programming. But it felt like time well spent.

Making Screencasts

The very first screencast I worked on was not actually part of the Physics 4A curriculum at all. I chose to do *An Introduction to Trigonometry* because it is something that is very straightforward and that my Physics 2AG colleagues could eventually use, and doing so would involve trying a

* To be fair, I wisely chose to somewhat limit what simulations I put my mind to coding!

variety of techniques and solving a variety of technical challenges. I spent many weeks trying and discarding various approaches. As I got more facile at using *Doceri* (the screencasting app I ended up choosing) and at using animations in PowerPoint, things got quicker.

The first 4A lectures I put together, for the second day of the 4A course, cover the topics of Position, Velocity, and Acceleration. This is about 55 minutes of so of video all together. It took a total of a month of back and forth to learn the tools and figure out the techniques well enough (and to write the VPython simulations) to produce and post that one class day's worth of videos. As I got better at this, things got quicker. I could create, draw, script and record ten to fifteen minutes of video for a relatively simple topic (relative motion) in perhaps 20-25 hours. More complicated screencasts involved more drawing, some programming, and using both PowerPoint and Doceri to import movies and animations and took more time.

Dead ends and tangents

I tried and abandoned a lot of ideas. For example:

In the process of putting the first set of videos on Position, Velocity, and Acceleration I found video of a recent SpaceX rocket launch, analyzed the video frame by frame to get altitude and speed data. I found the rocket's specifications for initial mass, thrust, and fuel burn rate. I plotted the data from the launch video, and tried to use that and the rocket mass and thrust specifications turn it into an activity we could use in class on relating position, velocity, force, and acceleration. I turned out that the data (from a curving, 3-dimensional rocket launch) did not fit nicely into a one-dimensional model that we could model. I would have needed the three dimensional position and direction data to get the model to work and I didn't have that data.

Another project involved using classroom handout developed by a high school teach using NASA-provided data for the accelerometer readings from a Saturn 5 rocket launch. I worked through the data and got results that didn't match the handout. It turned out that the instructor had made some "unphysical" assumptions about the data and fudged a number in order to get the right results.

Just pursuing these two projects (and a variety of other activities on the same NASA teacher resources webpage) took three full days. None of that work shows up in what my students see in the classroom videos.

I learned about the online course I took in Behavioral Economics from an email from edX. The course looked interesting. I focused a complete week and a half to work through it; bought the book the course was based upon, and found that the ideas in course ultimately colored the approach I took in structuring my flipped classroom.

The Value of a Sabbatical

Real learning is an inefficient, time-consuming process.
I pursue a revolutionary transformation of my classroom teaching approach. I learned a programming language and created unique simulations of physical systems. I tried a lot of ideas and approaches that did not make it into the final form of what my students see or experience. And I "failed" at a lot of things before I got them right, and got good at some things that I ended up not using.

I learned and accomplished the vast majority of what I set out to learn and do. And much of what I learned was outside of the immediate purview of my sabbatical proposal. Whatever "Measurable Objectives" or "Outcomes" my sabbatical was designed to produce, the actual learning I accomplished is much richer than whatever specific items is on that original list. This experience reminds me that students learn a lot in our classes, and some (or much) or what they learn may lie outside of what is listed in the official course outline of record.

The most valuable aspect of having a sabbatical is the uninterrupted time and mental space to single-mindedly pursue a particular objective and to simultaneously discover, delight in and examine fascinating tangents and pathways that only make themselves visible when one has the freedom to follow them. I approach the new school year invigorated and with a delightful sense of anticipation of how all of what I have learned, intended and unintended, will bear fruit in the coming semesters.

What it takes to flip a classroom—my conclusions and advice.

The simple answer is <u>TIME</u>, and lots of it. It took me, on average, two to three hours for each minute of posted video that I produced. This includes scripting, drawing or creating slides, writing simulations, recording, editing, and posting the final video.

It also involves investing in some technology. At a minimum this included:

- an iPad
- a stylus
- a headset microphone (I should have spent more on this).
- Doceri (software)

It also requires a quiet place to record the voiceovers for the videos.

Had I not had this sabbatical, I would not have had the time and freedom to focus on producing materials and figuring out the mechanics of how to flip this course. No one during the course of the regular semester has 30 hours per week to produce ten minutes of video or to explore all of the options I explored in order that total down to 30 hours. Part of the value to the college of my sabbatical is that I can streamline the path for anyone else who wants to make and post videos for their student to watch.

Even when the course videos are made, there will still be student video notebooks to read and check and quizzes to write and grade. Very likely running a flipped classroom will not save me any teaching time, at least at first—it will just change the focus from lecturing to managing more activities in my classroom.

Implementing a flipped classroom is made much easier by my department's Physics and Physical Science courses being taught in an integrated lecture/lab format. Other departments could create the same thing by linking a section of lecture with a section of lab and having the same instructor for both sections.

If a *department* made a decision to flip a course, I could imagine a collective project where one afternoon each week the department gathered to storyboard, draw, and script a lecture topic. One of more members could program simulations. Another could seek out appropriate images on the web. Yet another could draw the figures and another could do the voiceovers and recording. But it would take a focused and dedicated department to choose to add tens or hundreds of person hours on top of their existing teaching load to make such a large transformation in their teaching approach.

I am excited by the possibilities for student success and engagement that a flipped classroom offers. And as much as I believe that flipping a classroom is the way to go, I am cognizant that just because it is a good idea doesn't mean that it is going to happen. Most professors do not have both the time *and* commitment to create these kinds of materials.

The time that goes into creating *one* flipped lesson needs to be weighed against the time spent teaching the same less over and over again each semester, in multiple sections, over a period of years. Certainly part-time instructors are unlikely to make this kind of commitment to a course or to the College. In the long run, creating flipped courses will probably be a more efficient and effective use of an instructor's time, but that much of that time has to be invested on the front end.

Based on my Behavioral Economics course . . . If the College wanted to go in the direction of teaching in a flipped format, it would need to incentivize this process—provide stipends (just like the old FIG grants used to do) and reassigned time for groups of instructors to create screencasts. (Alternatively, the stipend could fund work during a Winter or Summer intersession). Just providing money is not enough. Money does not create the kind of time necessary to flip a course.

On the technical side, the College should make the video production side of this as easy as possible by providing quality microphones (and perhaps iPads) and a designated quiet space for making recordings, a designated studio for recording live video lectures (if that was the approach instructors wanted to take), a videographer to capture video in a classroom over the course of a semester and paid, designated experts with skills in graphics software to create images or animations.

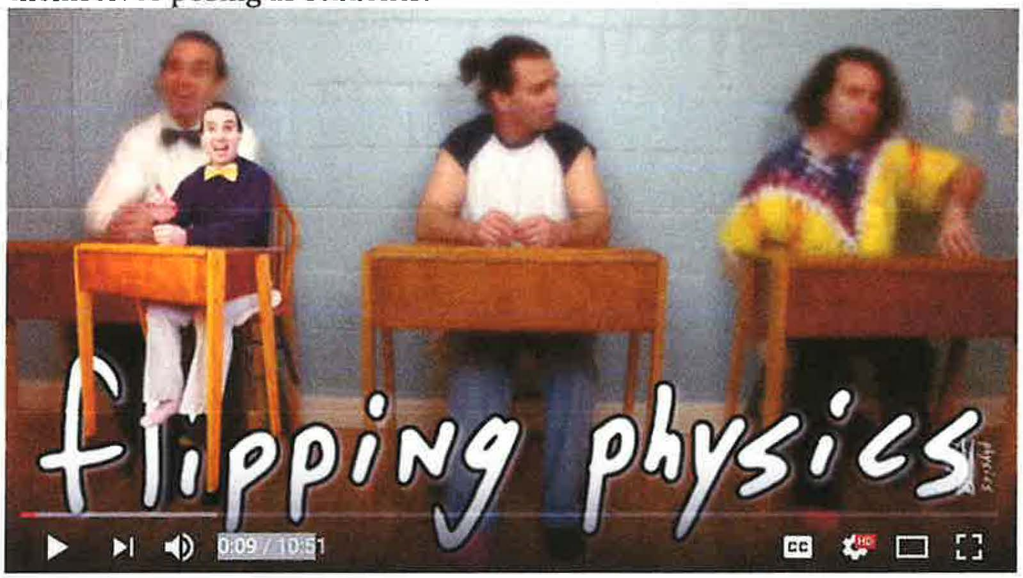
What comes next...

At the end of the Fall 2016 semester I will put together a presentation for my department and go through all of the materials I have created and what my experiences have been. If they find value in it, the department can assimilate these materials into a variety of courses and we can consider making addition screencasts or simulations for other courses beyond Physics 4A.

In addition, I will be giving a Flex Day presentation at the start of the Spring Semester on how to do screencasting and will share my experiences with a flipped classroom and what I have learned.

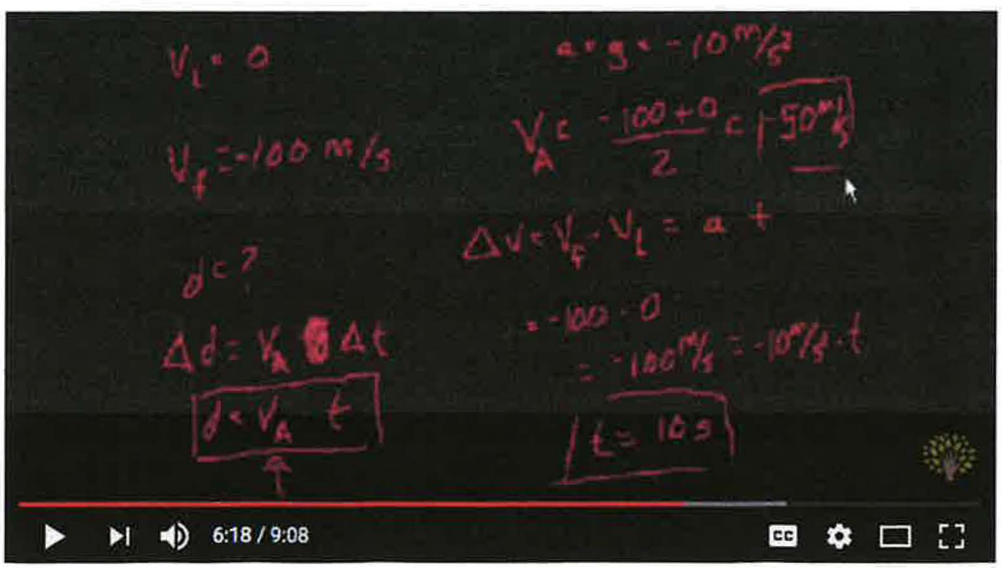
Presentation format/creation platform decisions

If you peruse YouTube for Physics screencasts you will find a lot of them! Some are technically sophisticated, in which instructors use green-screen techniques to position themselves in fantastical backdrops or to have three differently dressed duplicates of themselves posing as students:



From https://www.youtube.com/watch?v=q8TbXJEbeFg

On the other end of the spectrum is Khan Academy, which involves an unscripted presentation of someone talking as they screencast what they write on a tablet. It is sloppy, includes stuff crossed out, is done in one take and is presented without editing:



https://www.youtube.com/watch?v=15zliAL4llE

In between these two extremes are videos of someone lecturing at a white board, or someone using a video camera to record what themselves writing on a whiteboard or piece of paper.

The first approach's videos are technically amazing and sometimes excellent and engaging while others are at times slow to get to the point. Some of them are essentially fancy ways to give a boring lecture, but online! The latter approach (scribbles on a tablet) are mostly sloppy, disorganized, and unprofessional. The middle ones get the job done, occasionally with great humor and simplicity but sometimes with a lot more talking than is necessary to get the point across.

I attended a half-day workshop on *Making Physics Videos* at the American Association of Physics Teachers meeting in January 2015 in San Diego. Among many things discussed, the presenter talked about choosing what you were going to wear and making sure that your hands were in good shape and that your fingernails were clean, what kind of camera to buy, how to set up your studio, and what kind of lighting to use. It became clear to me that making high-quality, live Physics videos could be an expensive and resource- and space-intensive operation.

I also watched a screencast series on Lynda.com on *How To Make A Screencast.* I spoke with a Mt. SAC colleague in the Arts Division, who showed me how one could use Adobe Illustrator and After Effects to make drawings and then animate them. I spent a week or so watching and working through Lynda.com videos on how to use *Adobe Illustrator.* I concluded after that week that *Adobe Illustrator* was something that would take a long time to master, and that *After Effects* could be another long time.

My goal was to make videos to replace in-class lectures. I wanted to get down to the business of making videos rather than first becoming an expert on a complex software package or spending thousands of dollars.
I made a set of decisions about what I was going to produce.

Decisions Made

1) I am not going to be in the videos.

These videos are not about me. My goal is that they will be useful to other instructors in their courses and in my own course for many years. I did not want student responses to them to be colored by what I look like or what I am wearing in the video. I did not want my students in future years to be distracted by how different I look "now" compared to when the videos were made.

So I decided to keep what I look like out of them—just my voice and information on a screen. This had the added advantage of greatly simplifying the technology, expense, and training required to make these screencasts/lectures.

2) The videos will be scripted.

I made the first screencast material on trigonometry and tried recording the first part of one of them without a script. It was awkward. I ended up saying "umm" and realizing that I could have said something more clearly, had to backtrack . . .. The result had all of the disadvantages of giving a lecture but without any of the feedback from students as to whether something made sense.

It also was an inefficient use of video "time". There were way too many words compared to the information that I was trying to deliver. The entire point of making the videos was to get the lectures "right" and to make things as clear as possible. I believe that students do not want to wade through five minutes of speech on a video to get to a minute and a half of content.

So I decided that sounding professional and focused and being efficient with students' time would require scripting the videos.

And in order to have the videos be closed-captioned I was going to need to produce a transcript anyway. *

3) Each video will be short.

The general consensus from the discussions I read on the web is that 5-7 minutes is about the maximum length of time in a video before students will nod off or disengage. So I mostly broke the videos into shorter chunks.

4) Students have to be accountable for watching the videos

My discussions with Victoria Bhavsar at the Faculty Center at Cal Poly Pomona and my email discussions with John Suchocki, who has produced a wealth of *Conceptual Chemistry* videos, is that students have to be accountable for watching the videos. That means that coming prepared to class has to be a significant part (at least 20%) of their grade.

What was made very clear to me from the Behavioral Economics course that I took online was that if I want a particular student behavior (watching the videos, working through the problems in them) I have to make that the default behavior, and incentivize it and make it part of the core structure of my course.

My original intent was to have little quizzes embedded in Moodlerooms—students would watch a video and then complete a multiple-choice question or some numerical exercise before proceeding to the next video. But Moodlerooms is very clunky about these things. It was extremely time0consuming and tedious to enter a single multiple-choice question, and a matching question with 12 questions mapped to one of four possible answers proved impossible.

---

* Actually, the folks at 3Cmediasolutions.com will caption them for you. They promise to be 95% accurate, which is what the law requires. But that means that they can be up to 5% *in*accurate! I'd prefer just to get it right and have the transcripts be useful and correct.

I opted instead for embedding questions in the videos and having students maintain a separate notebook of answers to these embedded questions or problems. I would collect and check these notebooks at the beginning of class each day and start each class meeting with a quiz whose question(s) are remarkably similar to what students had been asked to do in the videos.

5) **The format of the videos would be**:

A) Here's some information.

B) Here's a worked example problem.

C) Here's an example to try on your own. The solution will be given in the next video

D) Here's a problem to do on your own in your notebook, to be turned in at the beginning of the next class.

This decision is based in part on my experiences with the online Electrical Engineering course I took. (Its format included embedded multiple-choice questions, but that course is hosted on a platform designed for that kind of thing and which is maintained by specialists.)

I found this approach (try this on your own, we'll look at the solution in the next video) engaging and encouraging in that I knew that if I couldn't figure things out I'd see the solution soon, and that if I *did* figure things out I'd get confirmation of that right away. It also mirrored what I would try to do in lecture

6) The lecture "on the screen" should, when possible, mirror the lecture experience of text being written on the board. So I would use an iPad app called *Doceri*.

I wanted the students' experience of watching a "lecture" on the video to feel more like the experience of being in class. When I introduce an idea or solve a problem in class I work through the steps, one at a time, showing each step and narrating as I go. *Doceri* acts like a smart whiteboard—I use a stylus on an iPad, and *Doceri* records each stroke as I write out what would have appeared on the board in class, except neater and with picture and diagrams often appearing fully-formed rather than being drawn one stroke at a time.

Sometimes I chose to use *Powerpoint*, either because I wanted to put up a big chunk of text (and typewritten text is neater and quicker than my writing) or I wanted to include a movie clip or an animation (not possible in *Doceri*). For *Powerpoint* I would usually have sections of text appear one line or section at a time rather than as an entire slide of text all at once.

7) Material in the videos would be either introductory, information delivery, or "mechanical". High-level, complex problem solving would be done in class in a supportive small group setting.

My intention for my "flipped" classroom is that students would work out challenging problems in a cooperative, supportive environment. Having them beat their heads against very difficult problems in isolation is exactly what I am trying to avoid!

So some of the material in the videos I have prepared for my calculus-based Engineering Physics course would work equally well for Physics 2AG (our introductory, trigonometry-based course). The last video or two in each cluster introduces something one step beyond what students have seen in Physics 2AG. The *Position and Velocity* and *Acceleration* material included in the appendix is an example of that.

Some of the material is essentially the Engineering-Physics level lecture I would have delivered in class (except better!), but benefitting from cleaner flow, better use of color, better penmanship . . . The material itself is beyond a Physics 2AG student but is not conceptually too complicated. The *Rocket Science* material included in the Appendix is an example of that.

And some of the material is just tedious but necessary math. Students need to be able to use calculus to calculate centers of mass and moments of inertia and to do gravitational force calculation for extended objects (as opposed to treating everything as a point mass as they did in Physics 2AG.). Students figuring out how to use calculus in Physics problems is probably *better* done at home rather than in the classroom in that each student can take it at their own pace and work through what they need to do.

Technology choices

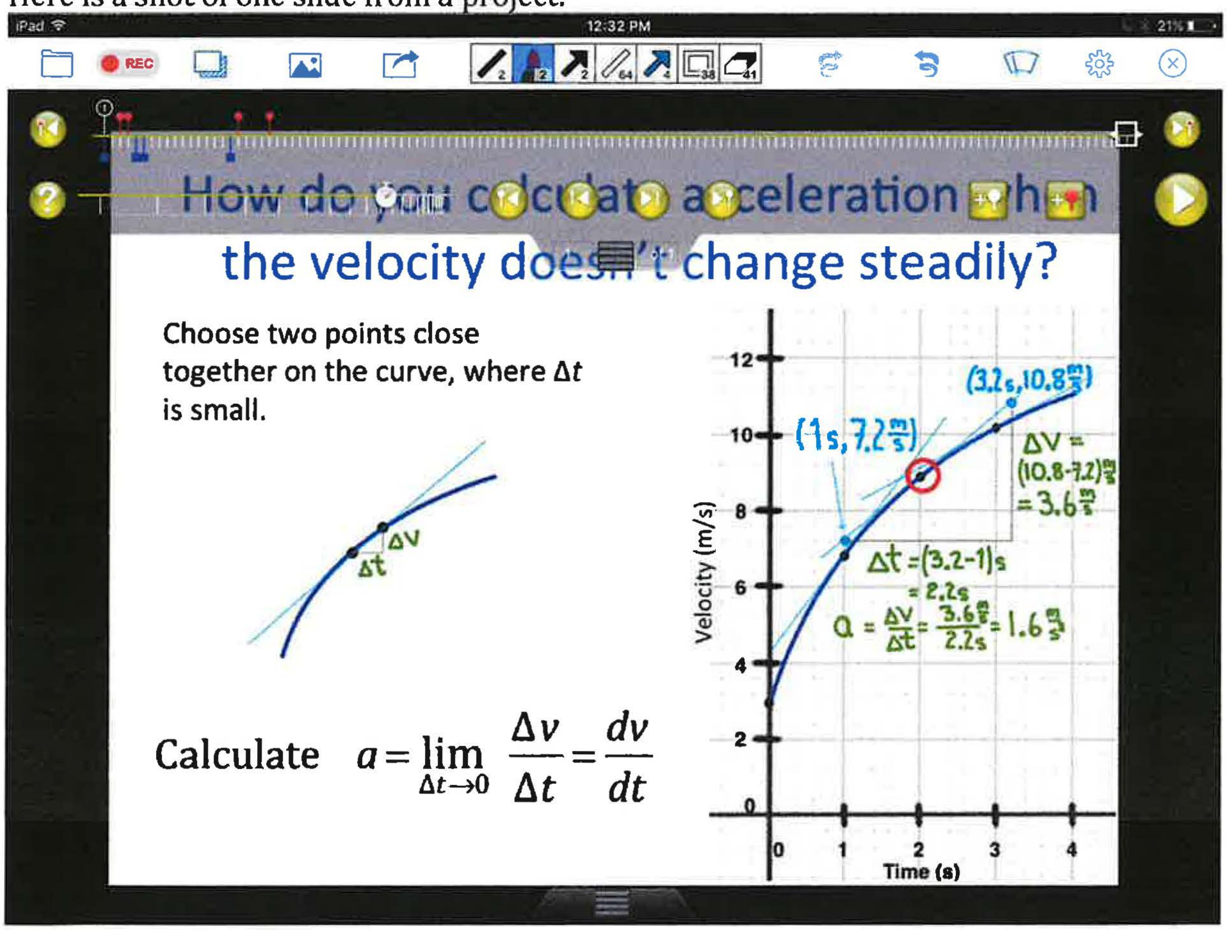
—How to make a movie/Making the movies available to students:

I made the decision to create my screencasts using Doceri, a piece of software that works on an iPad or Windows tablet and which allows the tablet to interact with a computer through a wireless network. Doceri also has the capability of recording whatever is on the tablet into a Quicktime movie.

The technology is relatively easy to learn. The app for the iPad is free, but linking it to a computer is $30, and that fee also includes an online training course on how to use Doceri. I bought a Jot stylus (~$30) that made drawing much easier than using my finger.

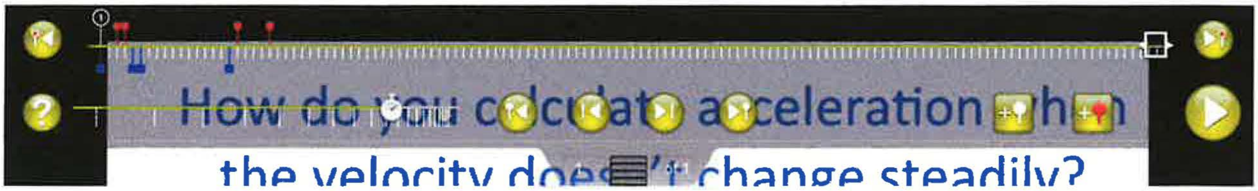
Each slide show/potential movie/presentation in Doceri is called a <u>project</u>.

Here is a shot of one slide from a project:



Along the top are various "buttons" which allow you to record a movie from the screen, change the background, insert a picture, upload the project, choose from a variety of pens and shapes, delete or undelete the most previous action, wipe the slide clean, or play with the settings.

Below that is a timeline (each white vertical mark is denotes one action):



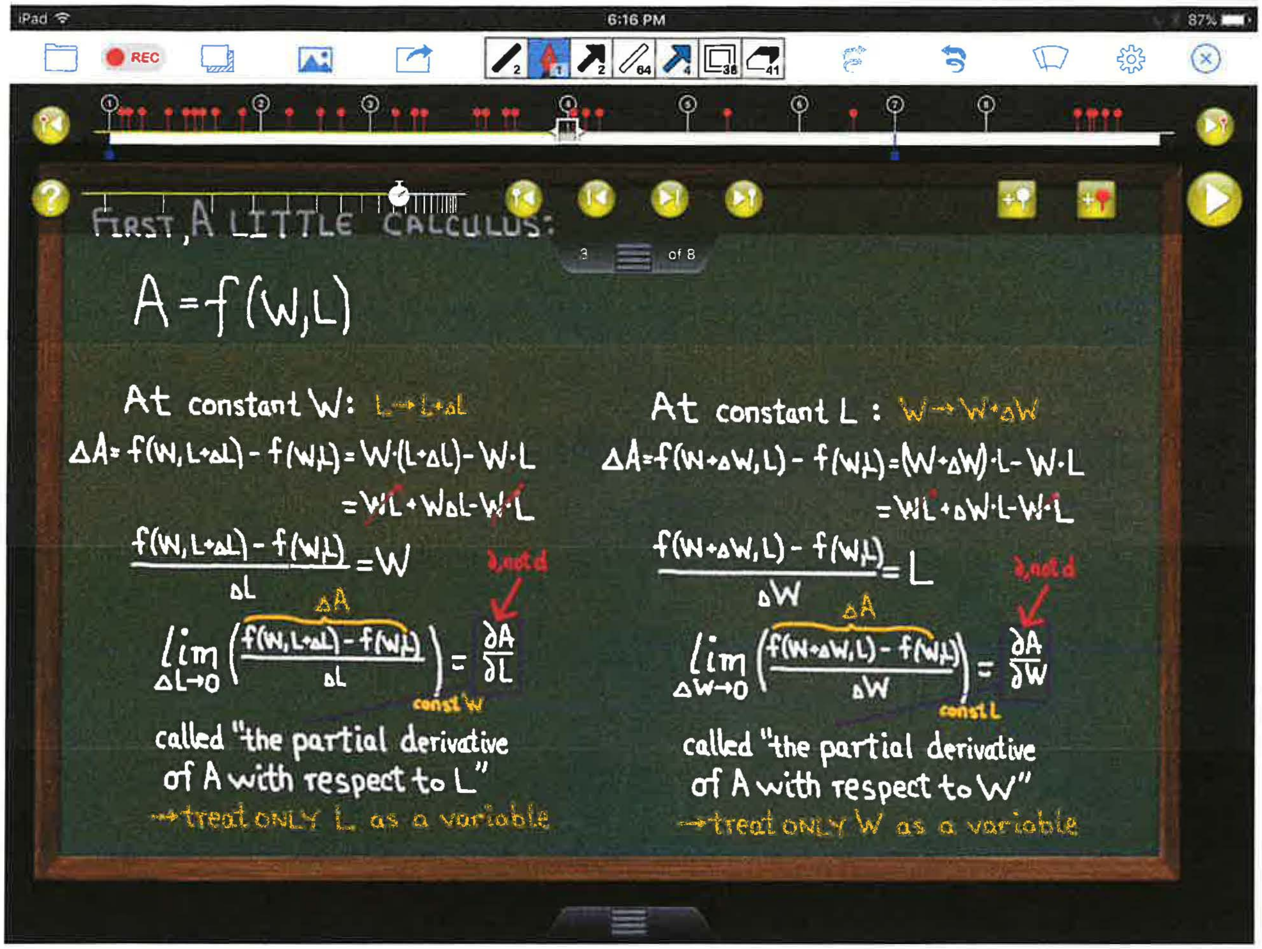
Each blue square indicates an inserted picture or image. Each red "lollipop" is a stop. There is a "stopwatch symbol" on a slider that lets you control how quickly the project plays back. The green circular keys in the middle let you advance or reverse the project one frame or an entire slide at a time.

The larger white and red "lollipop" +! buttons on the right allow you to insert a new slide or a stop, respectively. And the red lollipops on the upper right and left corners allow for advancing or reversing to the next or previous stop.

A "project" in Doceri consists of a set of actions. For the relatively simple slide above ("How to calculate acceleration etc."), for example, the actions were:

1. Make a Powerpoint slide with everything that was going to be text. (The graph axes themselves were drawn on a "graph paper" background in Doceri on the iPad, imported into Preview on my computer, given titles, data points, and a curve, then imported into Powerpoint.)
2. Then I took a screenshot of just the title and graph using the Doceri-computer connection and made it the background for the slide.
3. Then I took a screenshot of the complete Powerpoint slide with the next "build" (the Choose two points text). I cropped that in Photos, then imported that into the Doceri project.
4. Advanced powerpoint to the next build. Screen shot of the slide, crop, import into the project.
5. Drew dots, lines, and text in Doceri using the Circles, Lines, and Pen tools.
6. Re-cropped the original, complete Powerpoint slide to just include the "Calculate…" text and equation.
7. Drew Lines on the graph.
8. Wrote the rest of the text in on the Doceri slide by hand using the stylus.

Here is a slide that is only text, all written out by hand:



This is slide 3 of 8. Each red "lollipop" in the timeline is a stop. The pacing gets synchronized with a script so that each line on the "chalkboard" appears just as though I were writing it (neatly!) and saying it as I wrote.

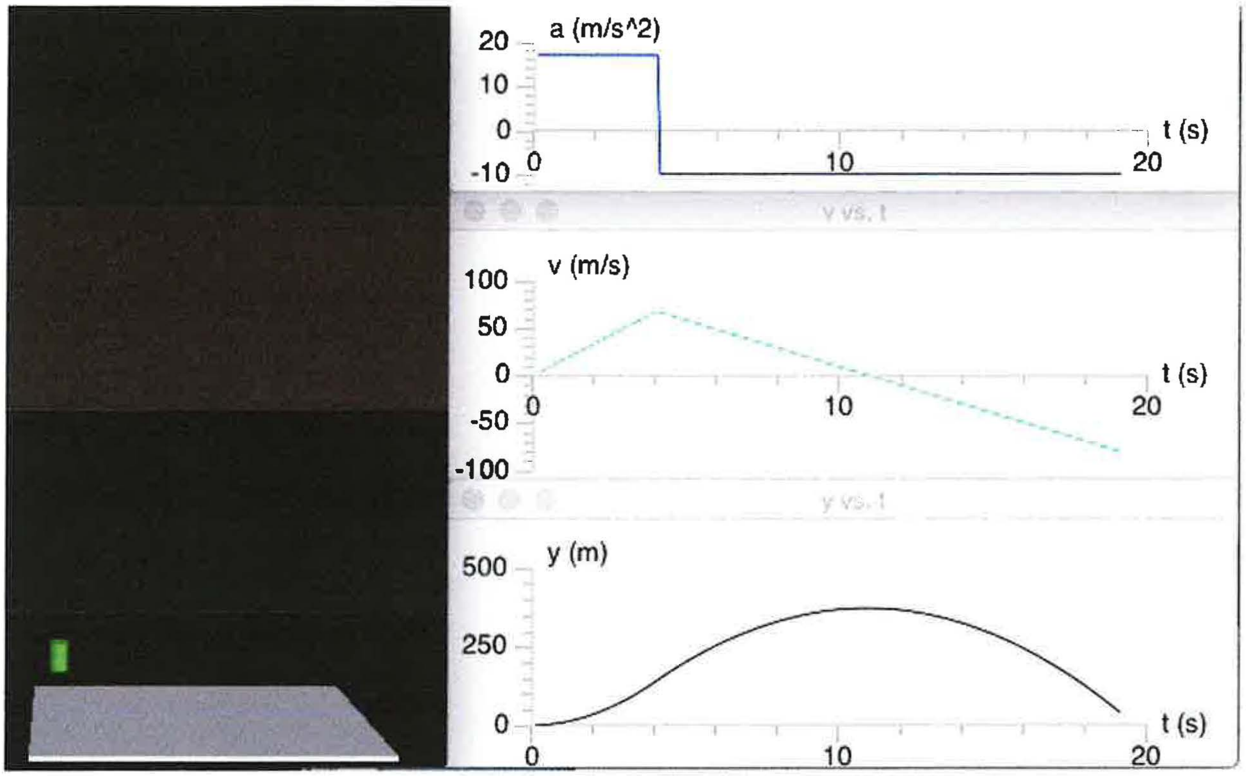
Doceri does have copy, cut, and paste functions, so I can duplicate and edit stuff that is similar to what I've already done (which is why the column on the right looks so much like the one on the left.) It *is* limited in that it records each stroke as an individual action, so one can't change the color or "font" of something that has already been written.

Because every stroke is recorded on a timeline, correcting a stroke means "rewinding" the presentation to where that stroke appears and then deleting and replacing the mistake. Otherwise, the original mistake appears and then the replacement happens wherever in time timeline the mistake was corrected.

Doceri can also be used to record a movie of whatever is one a screen. For a normal Powerpoint presentation, for example, one can just play through the presentation with whatever animations or builds and narrate through the whole thing, just as you would in class.

Here is an example of a VPython simulation of a rocket launch, which animates the launch (the cylinder on the left accelerates upward for four seconds, then runs out of fuel. It continues rising and slowing and then speeds up as it descends).

While this is going on, the graphs on the right show the acceleration, velocity, and height of the rocket in "real time" as the motion evolves. (I've included the code for this simulation as one of the examples of the VPython code section of this report.)



Here's what goes into producing this slide as a 25-second part of a video lecture:

1. Write the VPython program to animate the simulation and produce the graphs.
2. Run the VPython program, which produces the animation and graphs in a window on the screen.
3. Run QuickTime, selecting Capture Screen, and then selecting the section of the screen that I want to capture as a movie. click Start.
4. Run the VPython program again.
5. After the VPython animation is finished running, click Stop Recording.
6. Use QuickTime to cut off the beginning of the screen capture (so, all of the clicking and background that happens before the animation) and the end of the movie (same stuff after the animation is done).
7. Save the movie file.
8. Import the movie file into a Powerpoint slide.
9. Open the Doceri-computer connection over WiFi. Start recording in Doceri, then start running the movie in Powerpoint. Doceri captures whatever is going on on the computer screen.
10. Save the Doceri recording as a movie file.
11. Export that movie file back to the computer using Airdrop.

Recording a movie:

Recording a movie involves plugging a microphone into the iPad, hitting the "Record" button on the Doceri screen, and then starting to talk.

If what is being recorded is a Doceri project, you can tap through the various buttons (advance one stroke, play the project, which will just display the actions in sequence, much like a recording of what gets written on a smart board), or advance to the next stop (useful if you are starting a new slide and want to rewrite part of what was on the previous slide and use that as the point of departure).

If you are recording a Powerpoint presentation, one hits Record on the Doceri screen and then advances through the Powerpoint presentation on the computer while Doceri mirrors the computer screen. This is especially useful when there is a movie clip or animation you want to include in the presentation (Doceri can't embed movies or animations).

There are no editing capabilities in Doceri. If there is something wrong with a "take", you record all over again. I found it best to record in short time intervals (one to two minutes max) so that when I made a mistake and had to start all over again, what I had to re-record wouldn't be quite as long.

I uploaded each movie from Doceri on the iPad to the computer using Airdrop on the Mac.*
I opened each movie up, one at a time, in QuickTime Player, a free app on the Mac. I could only do simple things with this—split a movie clip, delete a section, or add one clip to the end of what is already open. I used this to "build up" a 2-10 minute movie from a series of shorter clips.

Each movie recording can involve recording on the iPad with the script on a laptop screen. (paper makes shuffling noises on the recording.) Or, if the iPad is recording from a Powerpoint presentation on the laptop, I had the script up on a second laptop.

Microphones:

The iPad has a built in microphone. I keep my iPad in a rugged case, which partially obscures the microphone and contributes an audible "white noise" hiss to recordings. I also tried using a small lavalier microphone I had gotten some years ago, a Logitech® microphone headset I borrowed from the OLSC, and an Andrea® headset with a "boomless" built-in microphone in the headset. The Logitech headset gave me the best sound, but all these still left an audible hiss in the recordings.

A Quiet Recording Space

One of the bigger challenges in producing the videos was finding a quiet space to do recordings. Working from home meant that recording could be interrupted by trucks going down the street, a neighbor jackhammering away some old concrete footings, dogs barking, phones ringing, my

* Airdrop is built into newer Mac computers and iOS devices. It allows one to wirelessly share files between devices. It requires a wireless network and Bluetooth. It is a very quick way to transfer files, even large ones like movies.

family wanting to use the microwave oven, or another neighbor's air conditioning clicking on and off, or myriad other sources of unwanted sound.

I tried recording in my office with the door shut, but people sometimes would have loud conversations in the hallway.

What worked best was:

1) Going to the radio station on campus and borrowing space on a part of a desk in a storeroom. That was pretty quiet.
2) Recording at home between 2 and 6 AM.
3) Recording in my office after 11 PM or on the weekend.

Uploading movies to a server

Community colleges instructors are fortunate that the Chancellor's office has funded 3Cmediasolutions (http://www.3cmediasolutions.org/). It provides server space for instructors to post movies and allows sharing of movies with others. It also allows you to upload a link to a movie on YouTube.

After applying for and receiving an account you can upload videos to your account and place them in folders. You give each movie a title and description, then upload it from your computer. It takes somewhere between ten minutes and several hours for the movies to become "Available" on their server, after which each movie is assigned a unique web address. You can also arrange movies into a playlist if you wish and send out a link to the entire playlist rather than to each individual movie.

Making the movie files available to students

Once the movies are posted to the 3Cmediasolutions server, I set up a "Book" in Moodlerooms. A book is a set of activities in Moodlerooms that students go through in a sequence, chapter, by chapter, where each chapter is a separate movie.

Here is an example of what that looks like:

Position and Velocity

Distance and displacement.

Going between position and velocity graphs.

A total of 8 videos.

Position and velocity videos

Script for Position and Velocity videos 87.1KB PDF document

Once a student clicks on the book (Position and Velocity videos, in this case) they get to the first chapter:

TABLE OF CONTENTS

1 Position and Velocity part 1
2 Position and Velocity part 2
3 Slope of a curve
4 Position and Velocity part 3
5 Position and Velocity part 4
6 Position and Velocity part 5
7 Position and Velocity part 6
8 Position and Velocity part 7

Position and velocity videos

◁ ▶

1 Position and Velocity part 1

https://www.3cmediasolutions.org/privid/51345?key=268b743d91d3e8bb88850217363b3ea63

◁ ▶

Clicking on the link takes them to the first movie. Clicking on the ◁▶ arrows advances to the next "chapter" in the book. One can also include readings, webpages, or whatever in a book.

Accessibility

State law requires that movies shown in class must be close captioned. But for movies made available asynchronously (so, online, where each student accesses the movie individually), the law allows you to provide a separate script for students to read from as the movie plays. So for each set of movies I included a script. Here is an example of a section of one script:

> Position and Velocity part 2
> Welcome back. We were looking at the distance and time data for a radio-controlled car in the lab. Our goal was to calculate the average speed for each of the three segments off the car's trip.
>
> The average speed is the slope of the position vs. time graph.
> For the first section of the graph we have
> V-average is delta x over delta t. Delta x is x final minus x initial, and delta t is t final minus t initial, so that's 1 meter minus 0 meters divided by 2 seconds minus 0 seconds, or 0.5 meters per second.
>
> For the second section of the graph we have
> delta x over delta t is 1 meter minus 1 meter divided by 3.5 seconds minus 2 seconds, or 0 meters per second.
>
> And for the last section of the graph we have
> delta x over delta t is 3.5 meters minus 1 meter divided by 6 seconds minus 3.5 seconds, for an average speed of 1.0 meters per second.
>
> +++
> Here is your trip from Santa Monica to the Ontario Airport, graphed again, but this time we are looking at your position at even smaller time intervals—in this case, every 0.1 hours.
>
> Your average speed for each of these 0.1 hour intervals is still delta x over delta t, where now delta t is 0.1 hr.
> If we recorded your position even *more* often the graph would start to look more like a curve.
> ...

Each blank line on the script corresponds to a pause in whatever is being written on the screen. Each +++ is a transition to a new slide.

Lecture videos

Each link listed below leads to a playlist consisting of a *set* of videos associated with each topic. Any given playlist typically includes all of the material students watch before coming to that particular day of class.

The first two playlists (*Trigonometry* and *Vectors and Vector Addition*) are introductory and most appropriate for the course before mine. I put them together first as practice and so that my colleagues might be able to use them.)

001 Trigonometry:
http://www.3cmediasolutions.org/f/ad59f37a0365bf1238ff61fefac4f4ccbe0cbb57

002 Vectors and Vector Addition:
http://www.3cmediasolutions.org/f/a3346e5e6b258c0dfed38f88ea9ac3127e7792d9

These next four playlists (*Position and Velocity* and *Velocity and Acceleration*) represent *one* night's worth of lectures, for the second day of class.
003 Position and Velocity:
http://www.3cmediasolutions.org/f/c28acb22523241c5969ec5ae9ebb2aca4366270e

004 Velocity and acceleration:

Acceleration Intro:
http://www.3cmediasolutions.org/f/59456529d46f458f6db17c97a515f2975f43eb4f

Constant Acceleration Kinematics:
http://www.3cmediasolutions.org/f/ba47824ff2fe11c4b2591d145f27efcde9b196ea

nonconstant Acceleration:
https://www.3cmediasolutions.org/privid/51403?key=54753a6b431cfe38ebbe943109ba14904359b288

005 Relative Motion:
http://www.3cmediasolutions.org/f/75950a27a707c88569a77d8fe6f1babb8fc088f5

006 Propagated Uncertainty
http://www.3cmediasolutions.org/f/cf6992770b40cead6dc0146476decac0b1594a73

007 Projectile Motion:
http://www.3cmediasolutions.org/f/30667fb880a103fd45d3a2baa085a7e1b92fa496

008 Newton's Laws:

Newton's Laws of Motion
http://www.3cmediasolutions.org/f/61c8b3b4ed9715ec83b9943fd6937309ad02cb22

Friction Forces:
http://www.3cmediasolutions.org/f/694dfc7454fb055350a771b54960623f831504b4

009 Vector Dot Products:
http://www.3cmediasolutions.org/f/e3e63ebb6f0df4a9418ddafe8ff383f9c5c565b5

010, 011 Quantity of Motion, Impulse-Momentum Theorem, Conservation of Momentum, Work-Kinetic Energy Theorem:
http://www.3cmediasolutions.org/f/f45e425b1c7644d65a0b4dfa59c3e5802e36c080

012 Rockets:
http://www.3cmediasolutions.org/f/4a3d2eca0d97f20ccfcc1d736dc072755f25f6fe

013 Calculus Areas and Volumes
http://www.3cmediasolutions.org/f/aca66868950aff6dd7747b0c2e73e9ff5d61d6c5

014: Center of Mass
http://www.3cmediasolutions.org/f/80c14dff7c45c5c01576b8d8cedae77d3c7ead43

015: Elastic Collisions, Displacements around the Center of Mass

Elastic Collisions:
http://www.3cmediasolutions.org/f/4bd54c70bc2dd813e32817b36959068cba20e334

Displacements around the Center of Mass:
https://www.3cmediasolutions.org/privid/55957?key=e0bc0fb05644b5142653ff9cccd411b58bd2ce29

016: Gravity and extended Bodies
http://www.3cmediasolutions.org/f/c7b232f811bcadf8b29d72948b71d86f32779f3a

017: Statics
http://www.3cmediasolutions.org/f/be98570fbf27d090af65fb6fe56038c761d6ab21

018, 019: Moment of Inertia, Parallel Axis Theorem, Rotational Dynamics Intro
http://www.3cmediasolutions.org/f/c90f0e6b5fe3a8ac81aec9a6d3063550db079104

020: Vector Cross Product
http://www.3cmediasolutions.org/f/851794a891fd10bf57f576078e7c9a5e91d1c21c

021: Introduction to Differential Equations; Simple Harmonic Motion
http://www.3cmediasolutions.org/f/5f28749a449bf3436747d6d40eec19bd2e8244b2

Position and Velocity

Position and Velocity Lecture videos—Script and Screenshots

The following pages contain the script for the video lectures introducing the topics of distance, displacement, speed and velocity. They were recorded into movies.

The "slides" themselves printed herein represent screenshots from Powerpoint (in which case they could involve animations that won't show up in a static screenshot) or Doceri (in which case the text and images on the screen appear as a sequence of pen strokes or images on the screen). Each screenshot included here is the *final* version of what appears on the screen right at the end of that particular slide. Although these give a sense of what the lecture looks like, best would be to watch the actual lectures themselves. (Links to those are included elsewhere in this report.)

Because I scripted the voiceovers, various words appear in the script to tell me when to advance to the next animation or set of strokes on the screen ("Tap" or Click"); how fast to set the rate at which things appear on the Doceri screen ("Speed 8"); or when to advance through a whole set of strokes all at once in Doceri instead of things appearing one stroke at a time ("Tab to next stop").

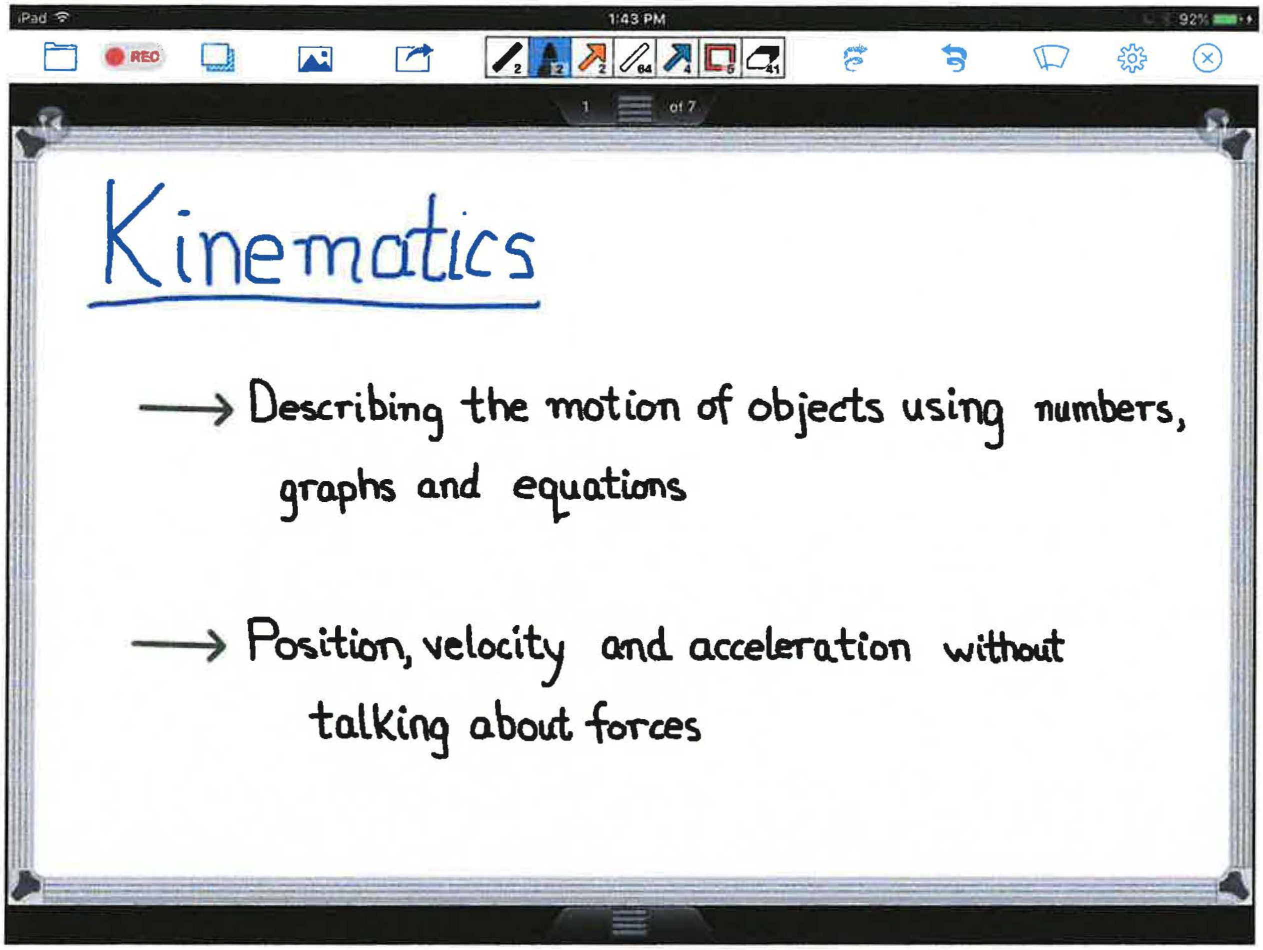
In addition, each lecture is a compilation of various slides, some from Powerpoint and some from Doceri, so there are notes at the beginning of some slides as to what presentation or Doceri project to start the narration from.

As a whole, this set of lecture videos comprises half of the first set of videos that students will watch in advance of the second day of class. This script is for seven videos, about 22 minutes total:

Title	Length (minutes:seconds)
kinematics part_1.mov	3:03
kinematics part_2.mov	2:07
kinematics part_3.mov	2:51
kinematics part_4.mov	2:04
kinematics part_5.mov	3:29
kinematics part_6.mov	5:12
kinematics part_7.mov	2:22

Links for students to access the videos are posted in Moodlerooms. I also post a student version of the script, which is essentially all of the text in what follows but without any of the "Tab" or "Click" instruction. The student version is separated out by lecture movie title rather than by the name of the source file.

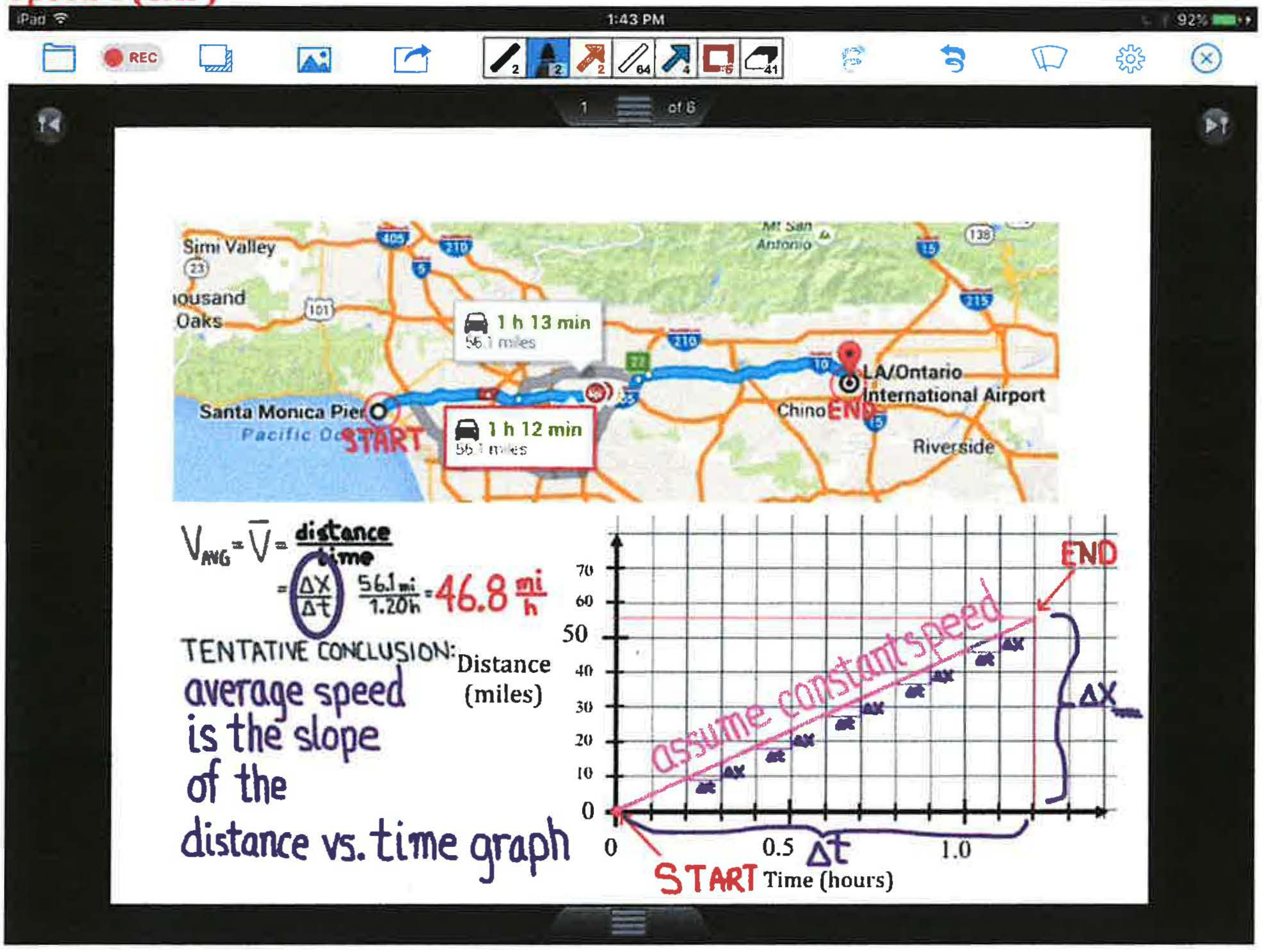
Kinematics slide 0



Greetings! Welcome.
The subject of the day is what's called kinematics.
This is a fancy word for describing the motions of objects using numbers , graphs and equations.
For us, this means we'll be discussing, position, distance, velocity, speed, acceleration and time without really talking about forces. Let's dive right in.

Kinematics slide 1

Speed 4 (TAP)



Imagine that you are at the Santa Monica Pier, and you're going to pick up a friend at the Ontario Airport.

And let's imagine that the trip is 56.1 miles and that it will take you an hour and 12 minutes, or 1.2 hours.

(TAP)

Your average speed for the trip is your distance traveled divided by the time: 56.1 miles divided by 1.2 hours, which gives 46.8 miles per hour.

(TAP)

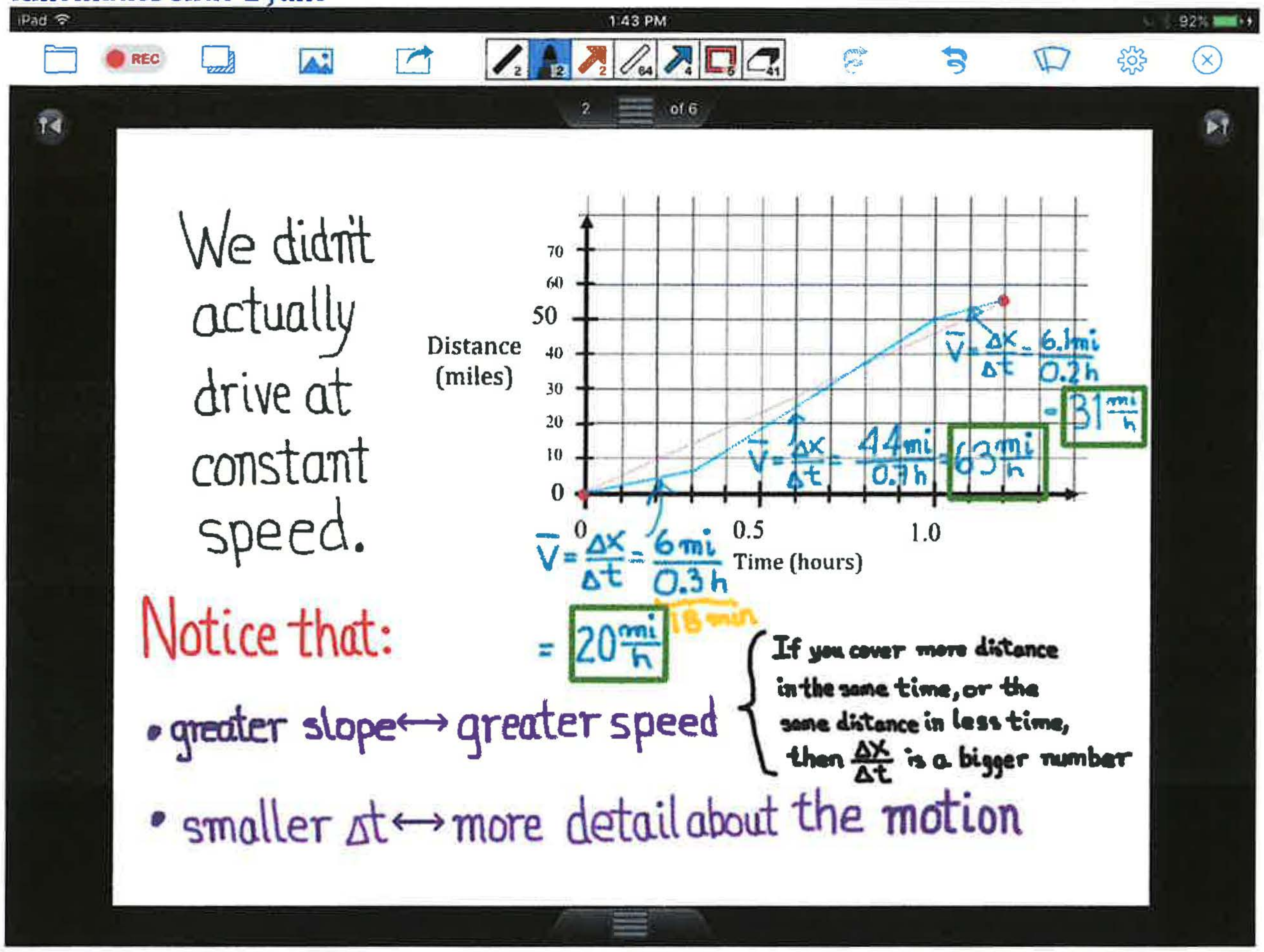
Let's imagine the simplest drive imaginable—you do the entire trip at constant speed. We'll plot a distance vs time graph, and plot our starting and ending points.

In this case the graph is a straight line joining the starting and ending points, because in each equal time interval delta t we travel the same distance delta x. For constant speed, the ratio delta x over delta t is the same for any time interval, or for the whole trip.

Speed 10 (TAP)

We can come to the tentative conclusion that the average speed is the slope of the distance vs. time graph.

Kinematics slide 2 June



In reality, you didn't do the drive from Santa Monica Pier to Ontario Airport at constant speed. You probably drove slower on the surface streets and faster on the freeway.

(TAP)

If we measure your distance traveled at a couple more points during your trip we get a more detailed description of your speed.

(TAP)

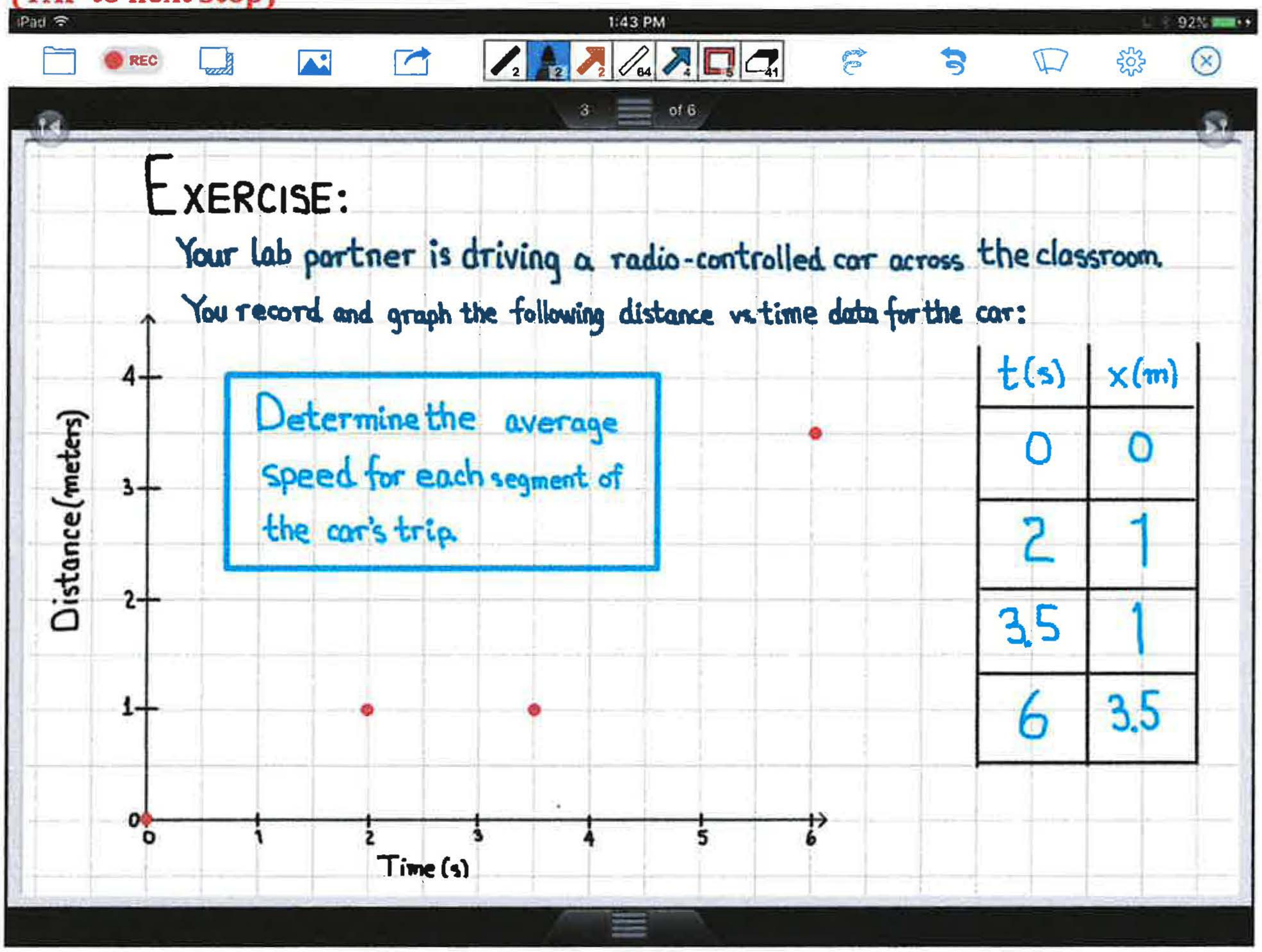
But notice that it's still true that a greater slope on your distance vs. time graph corresponds to a greater speed, because you're covering more distance in the same time interval, or the same distance in less time, so either way the ratio delta x divided by delta t is a larger number.

(TAP)

And again, notice that if we look at smaller time intervals we get more details about your speed during your trip

Slide 3

(TAP to next stop)



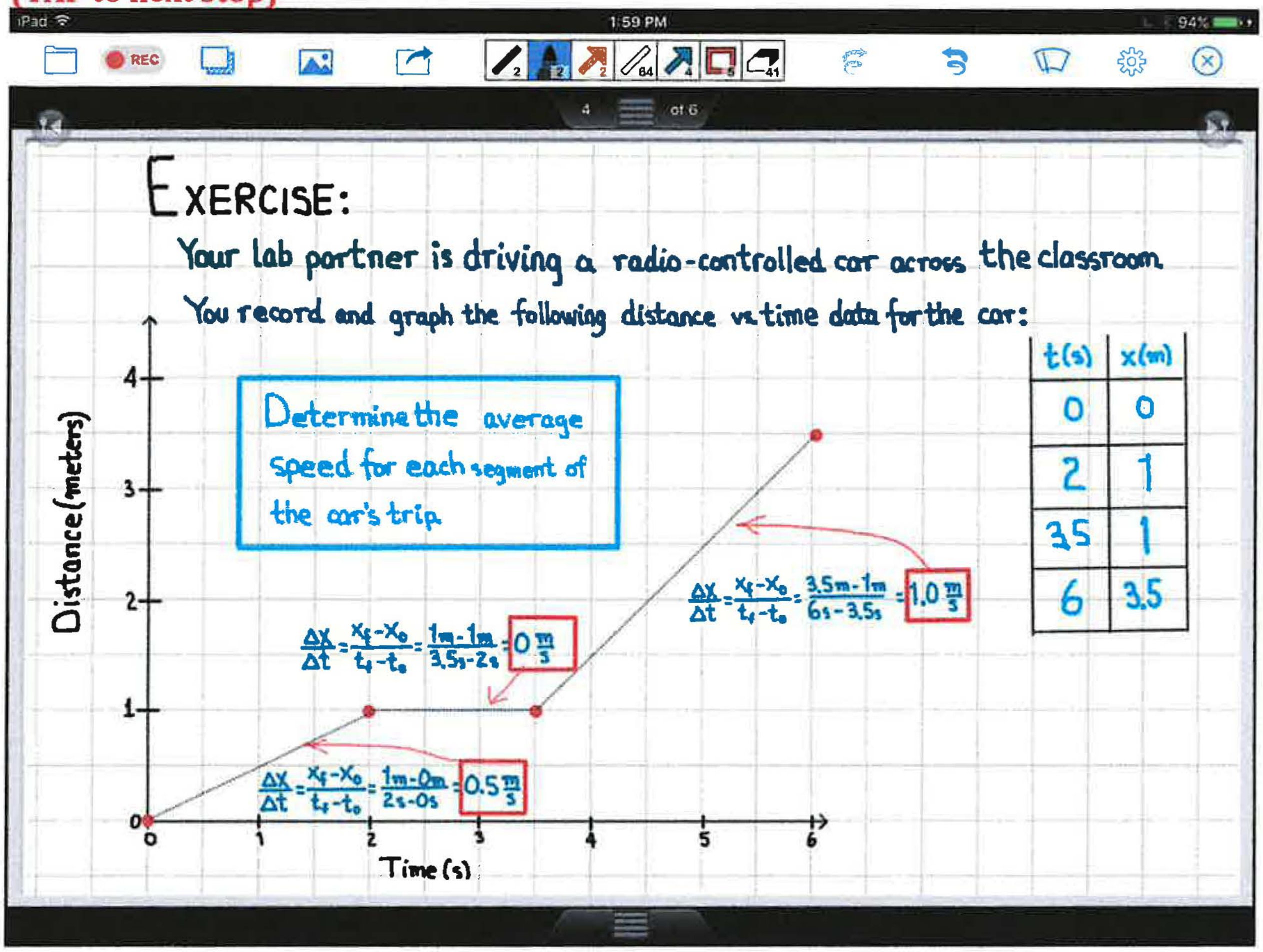
Here is an exercise for you to try on your own.

You are given some distance and time data, and are asked to determine the average speed for each part of the trip.

Work these out. I'll see you in the next segment.

Slide 4

(TAP to next stop)



Welcome back. We were looking at the distance and time data for a radio-controlled car in the lab. Our goal was to calculate the average speed for each of the three segments off the car's trip.

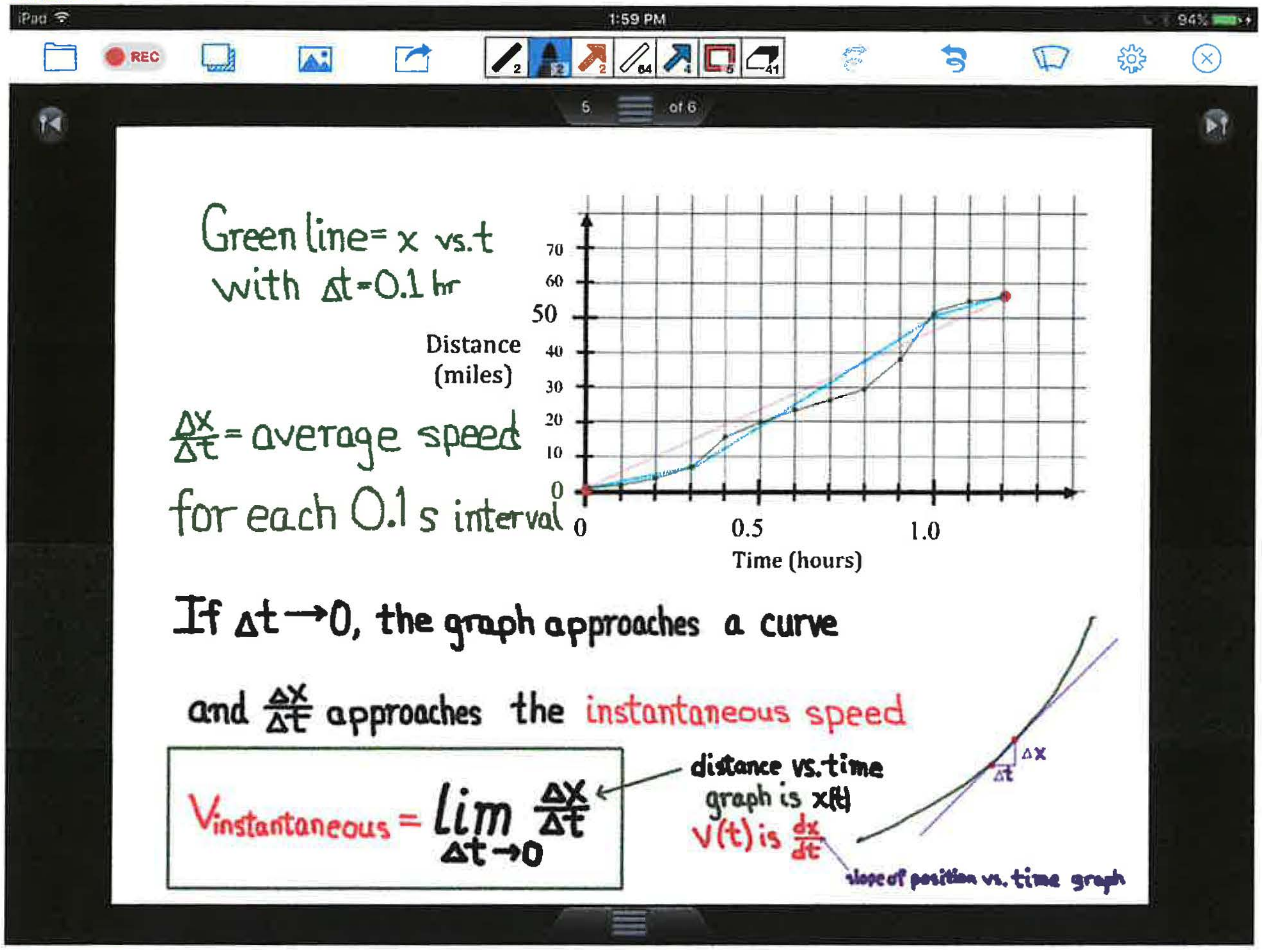
The average speed is the slope of the position vs. time graph.
For the first section of the graph we have
(TAP) V-average is delta x over delta t. Delta x is x final minus x initial, and delta t is t final minus t initial, so that's 1 meter minus 0 meters divided by 2 seconds minus 0 seconds, or 0.5 meters per second.

For the second section of the graph we have
(TAP) delta x over delta t is 1 meter minus 1 meter divided by 3.5 seconds minus 2 seconds, or 0 meters per second.

And for the last section of the graph we have
(TAP) delta x over delta t is 3.5 meters minus 1 meter divided by 6 seconds minus 3.5 seconds, for an average speed of 1.0 meters per second.

(Kinematics slide 3)

Slide 5



(Tap to first stop)

Here is your trip from Santa Monica to the Ontario Airport, graphed again, but this time we are looking at your position at even smaller time intervals—in this case, every 0.1 hours.

Your average speed for each of these 0.1 hour intervals is still delta x over delta t, where now delta t is 0.1 hr.

If we recorded your position even *more* often the graph would start to look more like a curve.

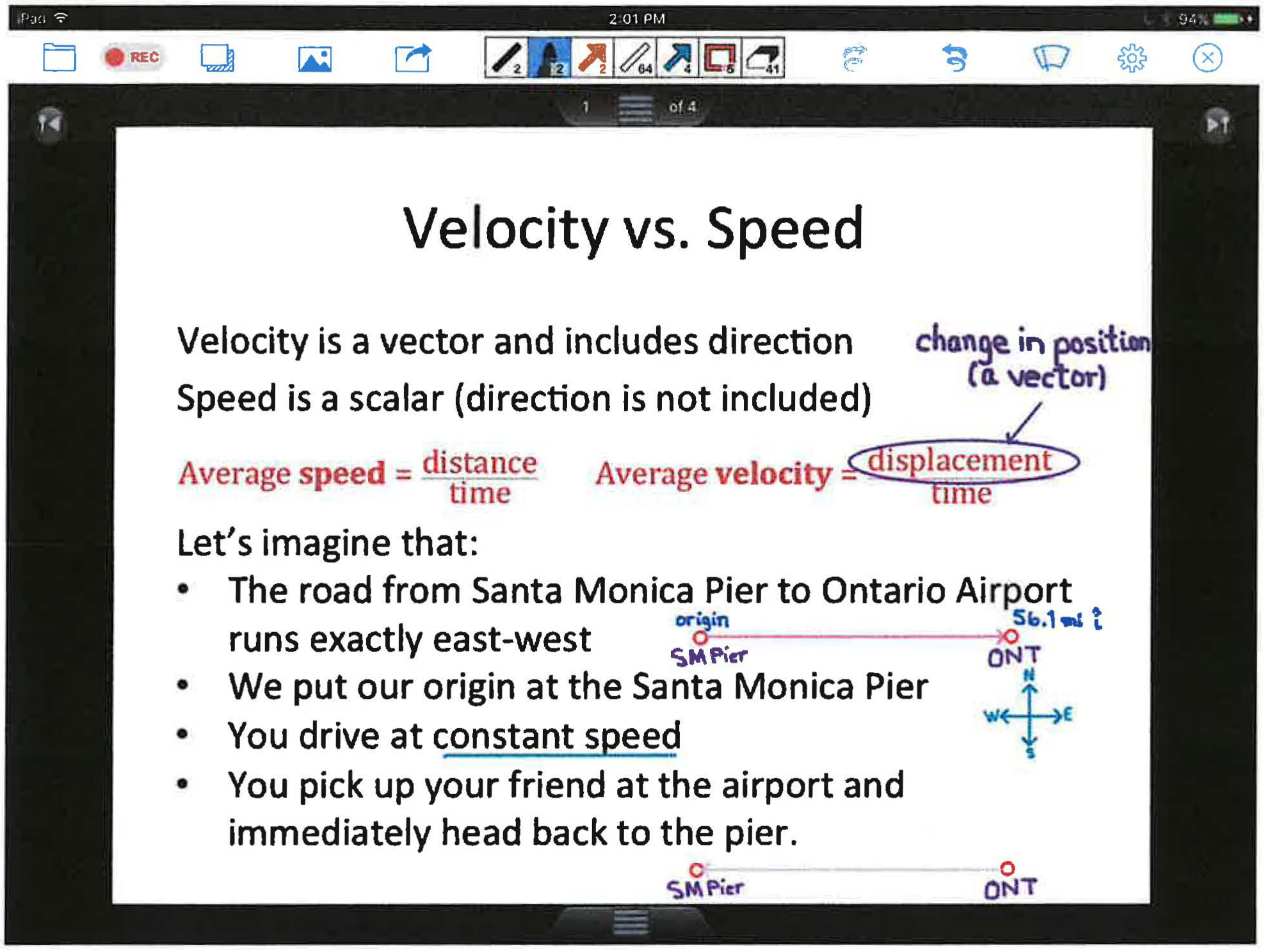
(TAP)

In the limit that the time interval delta t between distance measurements approaches zero, the graph actually *becomes* a curve, and the slope of the graph for each of these time intervals becomes the instantaneous speed for that interval. The interval is so small that we can talk about the speed at an instant (which is why we call it the instantaneous speed!) Here is a short video to illustrate this idea of taking the slope of a curve

Mech Universe Derivatives.Mov (2:13 minutes)

Kinematics vel vs speed rev 2

Slide 1



In Physics we make a distinction between velocity and speed. Velocity is a vector and includes direction, while speed is a scalar. Direction isn't part of speed.

We calculate average speed as total distance over total time.

(TAP)

We calculate average velocity as *displacement* (our change in position) over time. Let's look at an example.

(TAP)

Let's suppose that:

The road from Santa Monica pier to Ontario airport runs exactly east west, and that we put our origin at the Santa Monica Pier.

That means that our initial position is 0, and our final position (at the airport) is 56.1 miles times i-hat, the unit vector in the plus x direction.

(TAP)

Let's assume too that you drive at constant speed and that **(TAP)** you pick up your friend and immediately head back to the pier

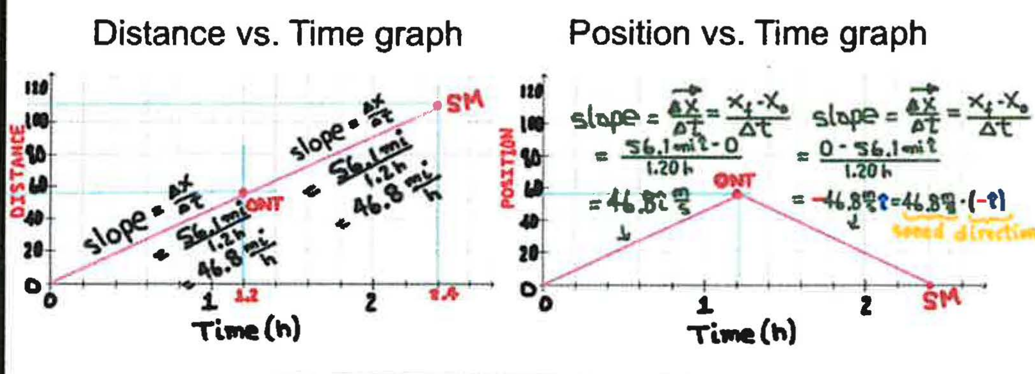
Let's look more in detail at this motion.

(TAP)

Slide 2

Velocity vs. Speed (continued)

Time (h)	Distance (mi)	Position(mi)
0	0	0
1.2	56.1	56.1 $\hat{i}$
2.4	112.2	0



(TAP)

You start at the origin (the Pier) and after 1.2 hours you have traveled at distance of 56.1 miles, and your position is 56.1 miles times i-hat, that is, 56.1 miles east of the origin. After another 1.2 hours the distance traveled is twice 56.1 miles, or 112.2 miles, but your position is 0—at the origin.

(TAP)

If we draw our distance vs. time, and position vs. time graphs, we see something interesting—the distance graph always goes up, because the distance always increases, regardless of what direction you're going.

In contrast, the position vs time graph reaches a maximum after 1.2 hours, when you arrive at the airport, 56.1 miles east of your starting point, and then goes back to zero when you arrive back at the pier.

(TAP)

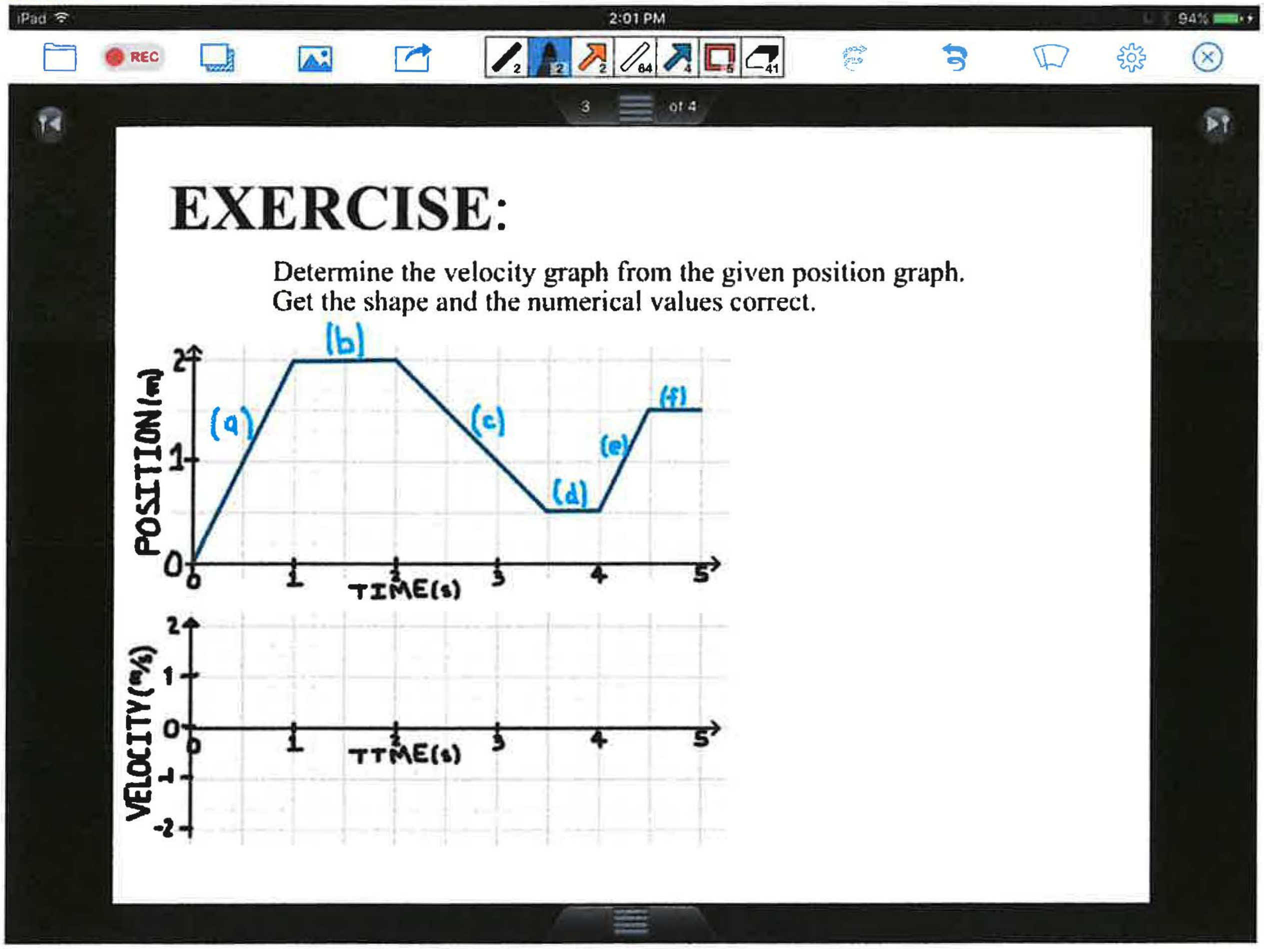
When we calculate the speed, we get a positive number because the distance is increasing with time the entire way.

(TAP)

But when we calculate the velocity, we see that it's positive for the first half of the trip (where your change in position is a positive number), and negative for the second half of the trip, where your change in position is a negative number.

(exercise v from x slide 1)

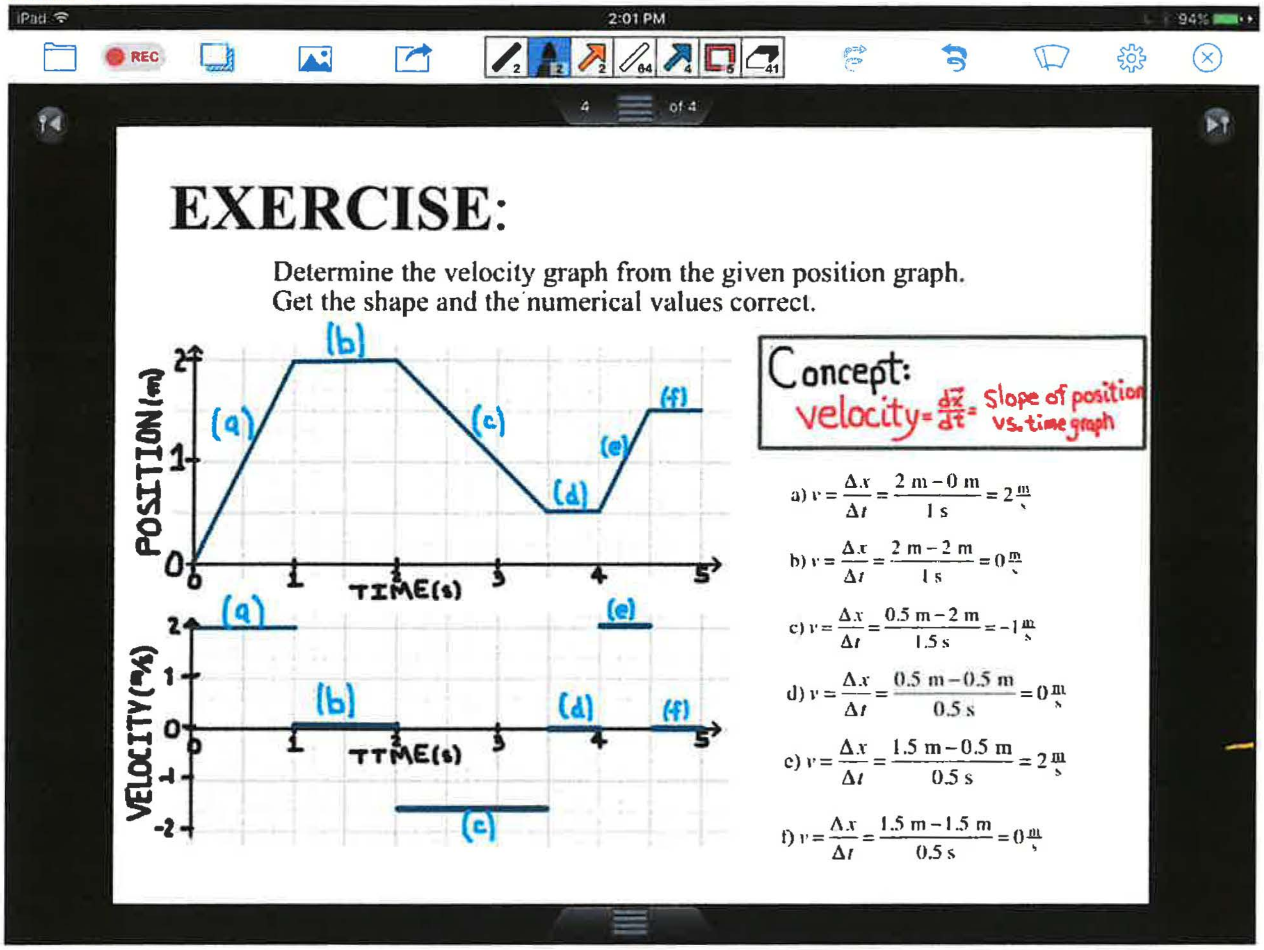
Slide 3



Try this exercise. You're give a position vs. time graph and you're asked to generate the matching velocity vs. time graph. Give that a shot and I'll see you in a few minutes in the next video.

(v from x solution)

Slide 4



Greetings! Welcome back.

We were looking at a position vs. time graph and trying to generate the corresponding velocity vs. time graph.

(TAP)

The concept involved is that the velocity is the slope of the position vs. time graph, change in x over change in t.

Let's look at the motion , one step at a time.

For step (a) you start at the origin and move 2 meters in the +x direction in 1 second.

(TAP)

Your velocity is your change in position divided by the time, or +2 meters divided by 1 second which gives a slope of 2 meters per second.

For step (b) you start and end at the same place, 2 meters in the +x direction from the origin

(TAP)

Your velocity is your change in position divided by the time. Since your change in position is zero, you velocity is zero. You can also see that the slope of the line for part (b) is zero.

(TAP)

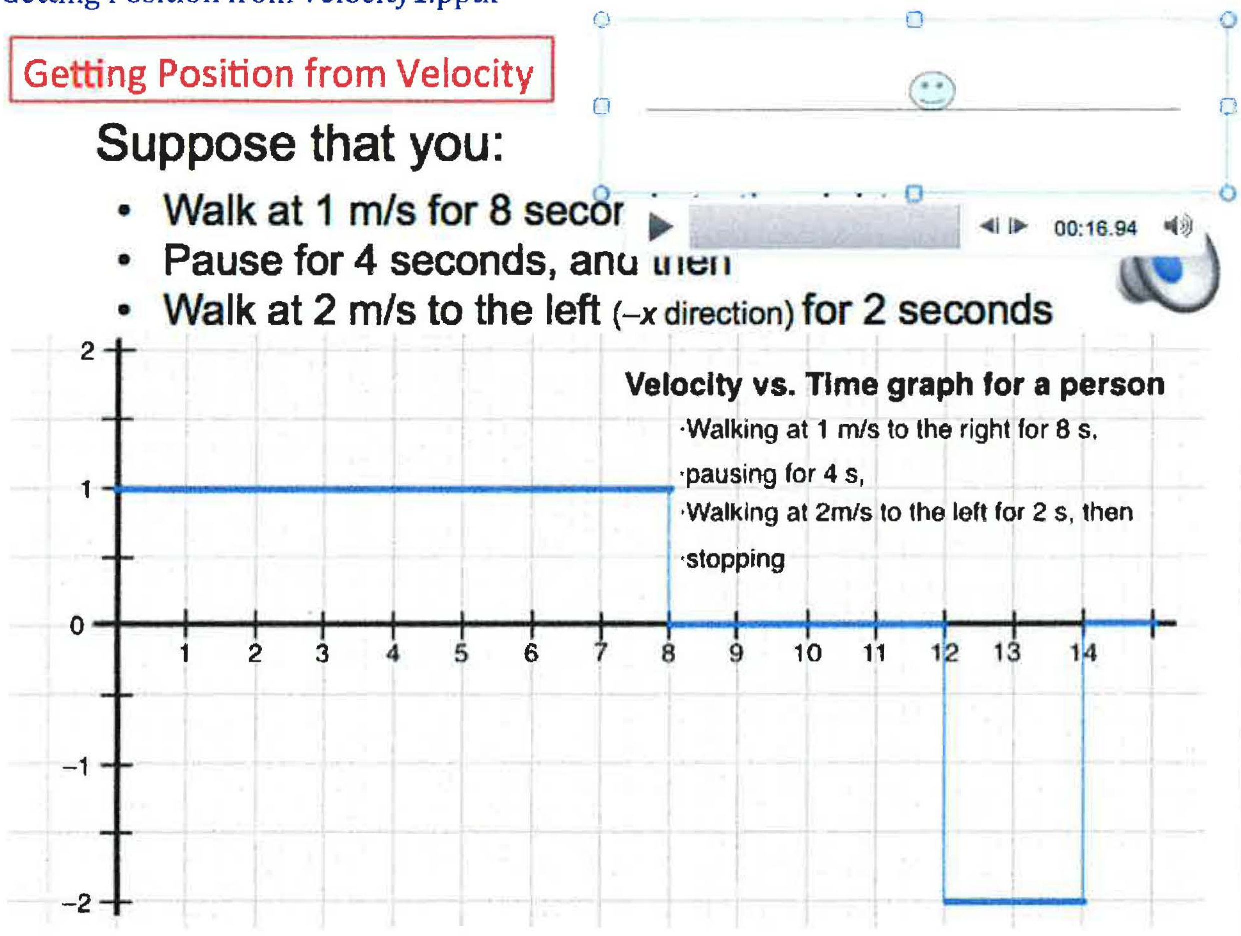
For step (c) you start 2 meters from the origin and end up 0.5 m from the origin. Your change in position is 1.5 m in the negative direction, or simply negative 1.5 meters. This displacement takes place over 1.5 seconds, so your velocity works out to be -1 m/s.

For step (d) you start and end at the same place, your change in position is zero, the slope of the position vs time graph is zero, and so your velocity is zero. **(TAP)**

For step (e) you start 0.5 meters from the origin and end up 1.5 m from the origin. **(TAP)** Your change in position is 1 m in the positive direction, and your displacement takes place over 0.5 seconds, so your velocity works out to be +2 m/s.

Finally, for step (f) you start and end at the same place, your change in position is zero, the slope of the position vs time graph is zero, and so your velocity is zero. **(TAP)**

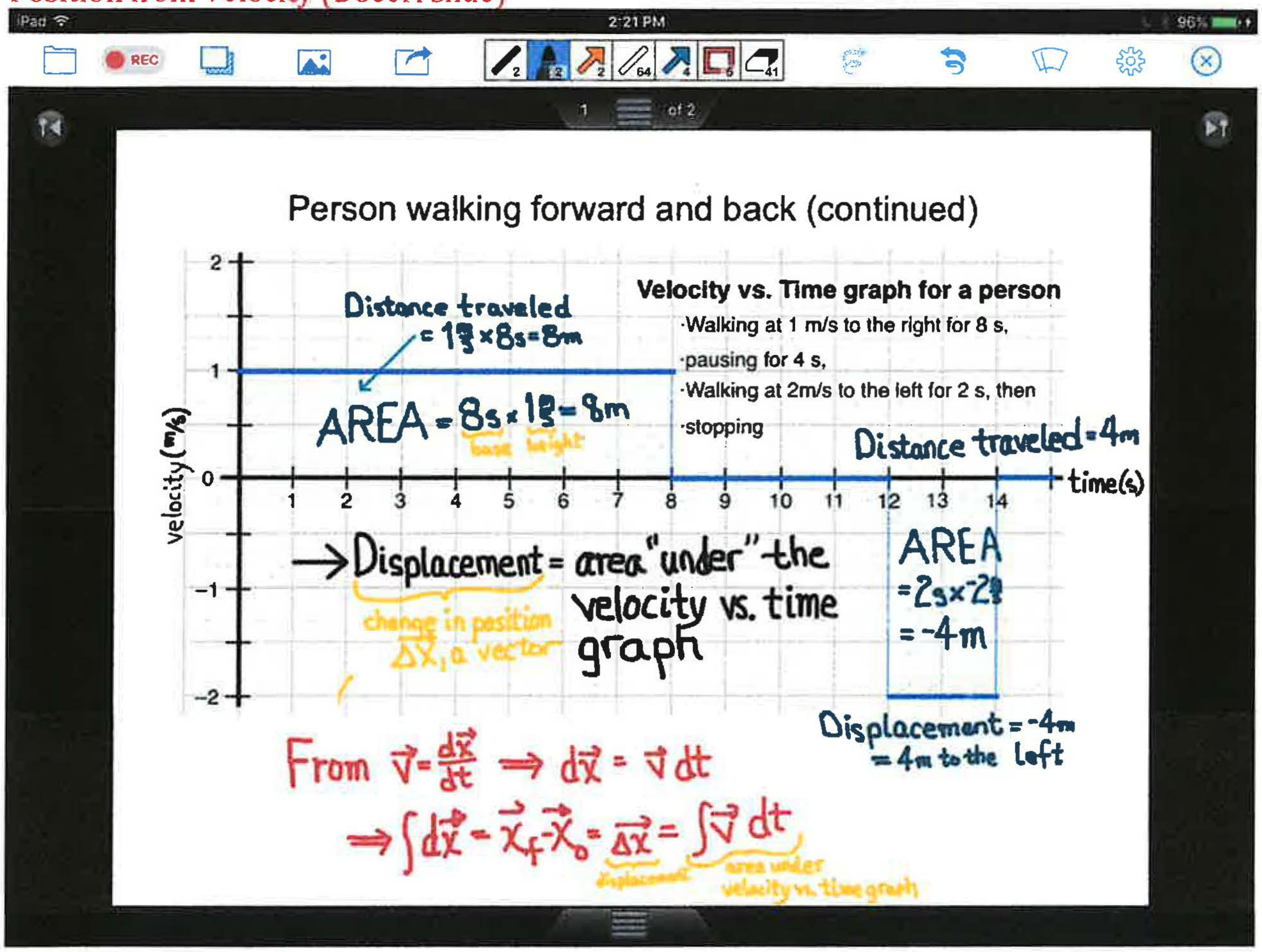
Getting Position from Velocity1.pptx



Greetings! Welcome back.
Now we'll look at the opposite process—getting the position graph if we know the velocity graph.
Let's look at an example where you walk in the plus x direction at 1 m/s for 8 seconds, you pause for four seconds, and then you walk in the negative x direction at 2 m/s for 2 seconds. The motion looks something like this (click movie) (click music)

The *graph* of the motion looks like this: **(click)**

Position from Velocity (Doceri slide)



Here's our graph again.

(TAP) Notice that in the first eight seconds you travel 8 meters—1 m/s for 8 seconds. This also happens to be the area under the graph—that is, the area between velocity line on the graph and the x-axis.

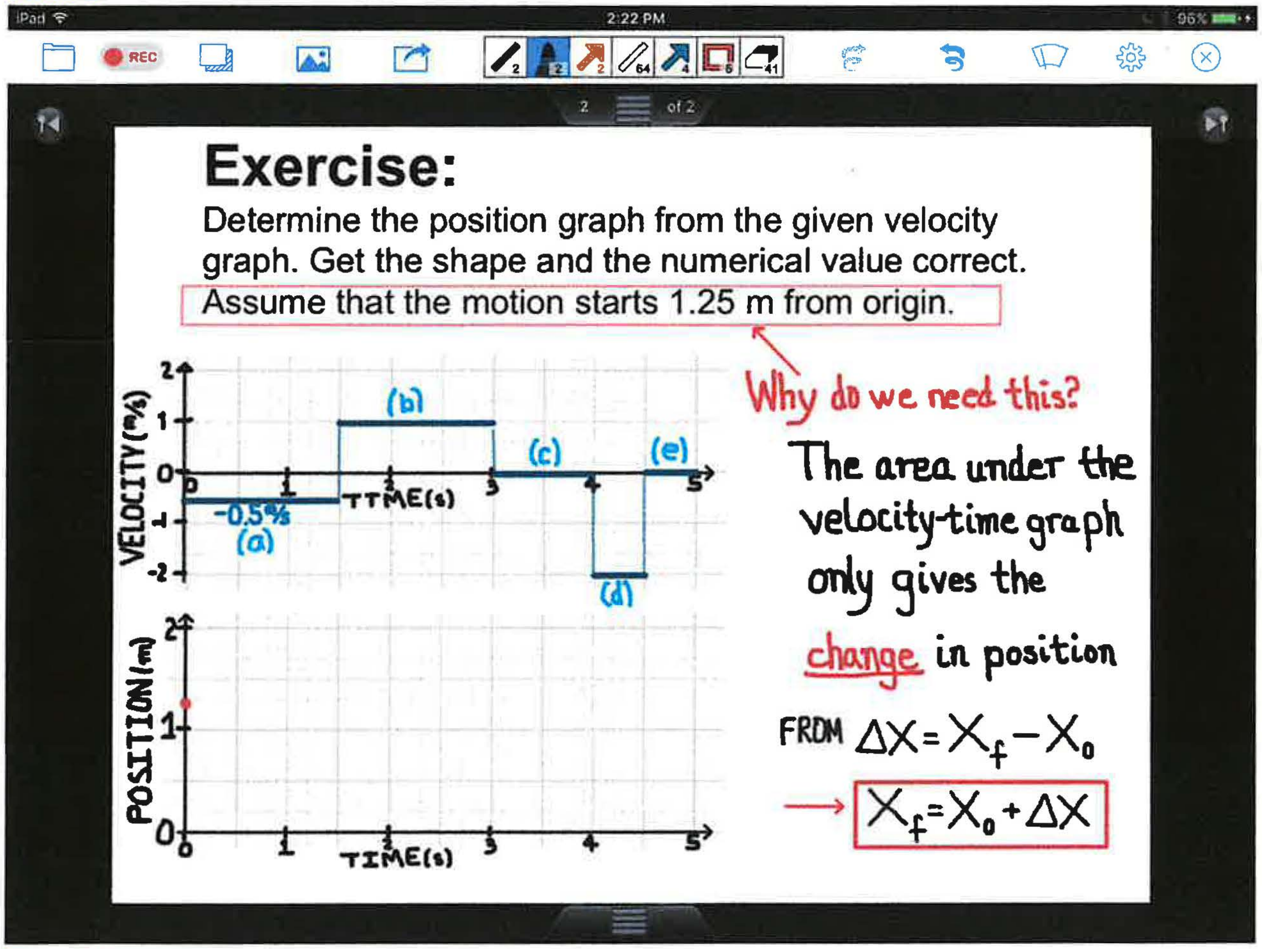
For the part where you are paused you don't go anywhere, so your distance is zero. Likewise, the area under the velocity line on the graph is zero.

(TAP) For the last part of your motion, you travel 4 meters—2 m/s for 2 seconds. But your *displacement* (change in position) is -4 meters because you are heading in the negative x direction. Notice that the area between the velocity line and the x axis here is -4 meters.

We usually call this area the area "under" the graph, but what we mean by "the area under the graph" is really the area between the graph and the x-axis.

(TAP) These examples tell us that we can find the displacement by taking the area under the velocity vs. time graph. This is what you already know from calculus
speed 4 (TAP) From v is dx/dt, we can multiply both sides by dt, then integrate both sides. Integrating dx gives the change in position, and integrating v dt gives us the area under the velocity vs. time graph

(x from v exercise rev2)
Slide 2

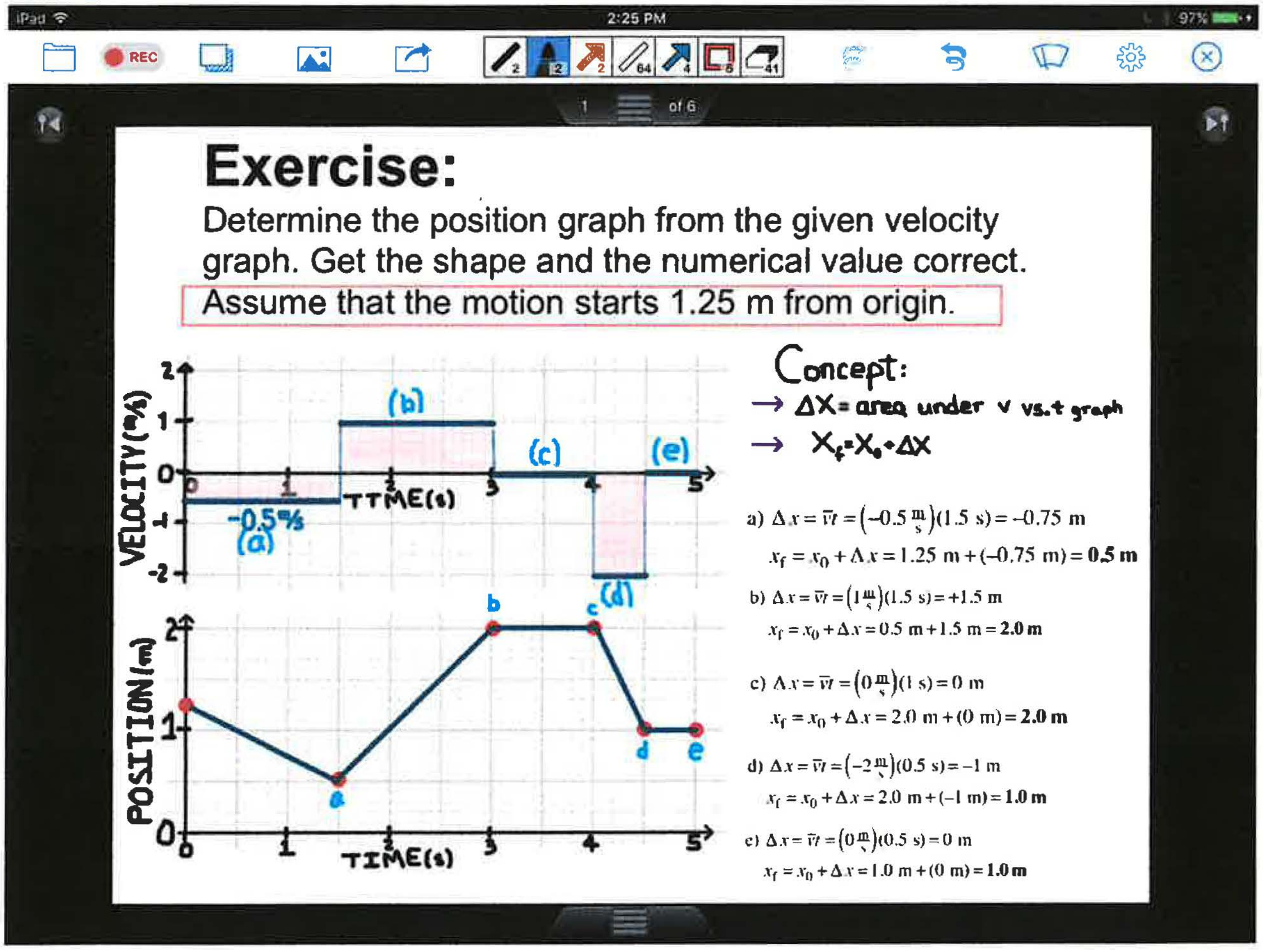


Tap t first stop
Here's a little exercise to check that you have this down. You are given a velocity vs. time graph and you are asked to generate the position vs. time graph.
You're also told that the motion starts at x equals 1.25 meters. Whey do we need *that* piece of information?
(TAP) The area under the velocity vs. time graph only gives you the *change* in position. You have to know where you started so that you know where you are changing position *from*. In math terms, x-final equals x-initial + your change in position.

So, try this out on your own and I'll see you in a few minutes.

x from v solution rev2

Slide 1



Tap to first stop

Greetings! Welcome back.

We were given a velocity vs. time graph and wanted to generate the position vs. time graph from it.

We were also given the initial position of 1.25 m, **(TAP)** so we'll put that starting point on our graph.

The concept we're applying is that the displacement is the area under the velocity vs. time graph. We can get the final position if we know the initial position (that is, where we started) and how much the position changed.

Speed 2

For the first part, the area is negative 0.5 m/s time 1.5 seconds, **(TAP)** which gives an area of negative 0.75 meters. Since we started at 1.25 m from the origin, this displacement of negative 0.75 meters puts us at 0.5 m at the end of step (a).

(TAP)

For part b the area under the graph is 1 m/s high and 1.5 seconds wide, for an area of 1.5 m.

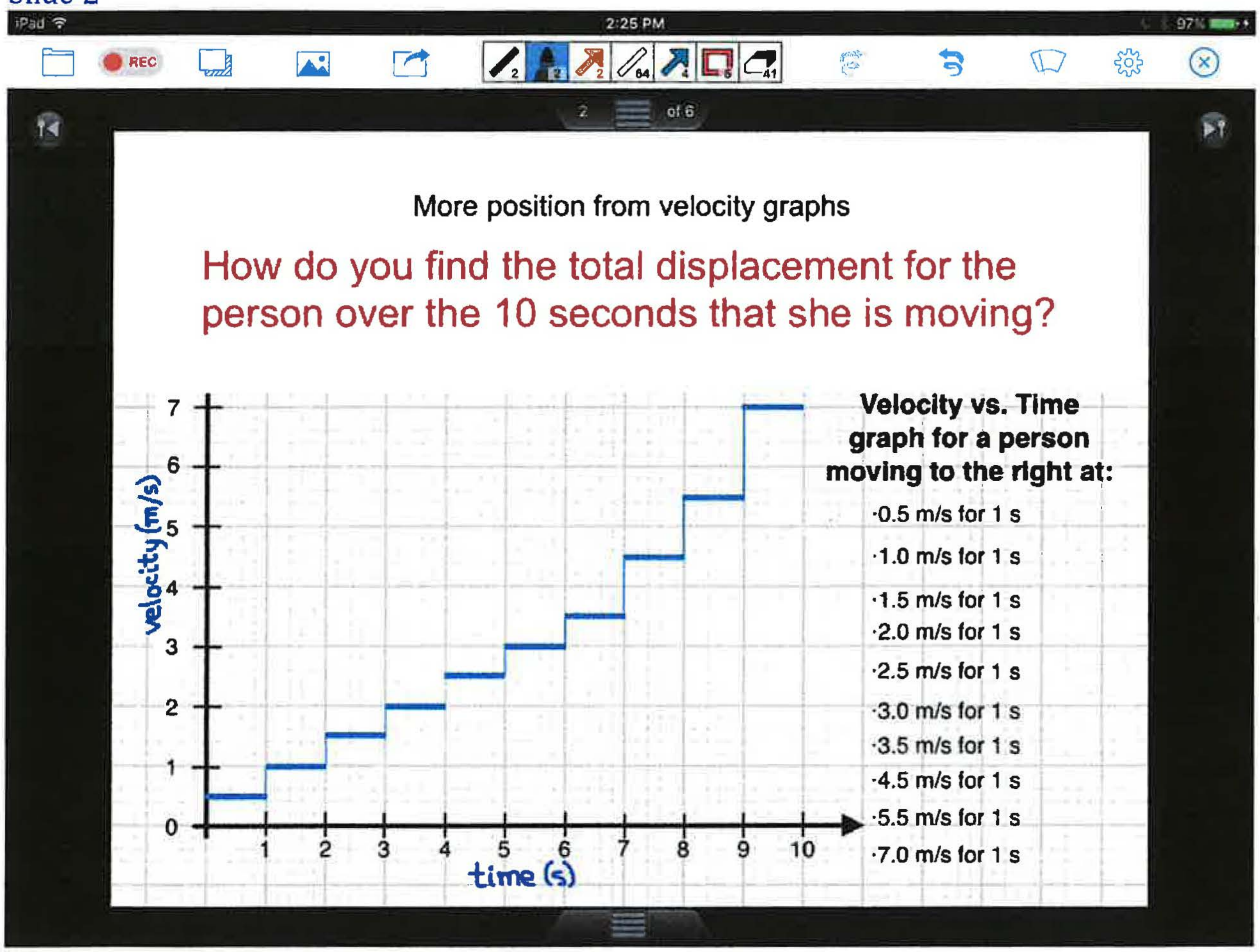
Since we were at x = 0.5 m, a displacement of +1.5 meters puts us 2 meters away from the origin

For part c the area under the graph is zero, so our position doesn't change. **(TAP)**

(TAP) For part d the area under the graph is -2 m/s high and 0.5 seconds wide, for an area of -1 m.
Since we were at x = 2 m, a displacement of -1 meters puts us 1 meters away from the origin

And finally, for part e the area under the graph is zero, so our position doesn't change. **(TAP)**

(Pos as sum of v vs t rects1)
Slide 2

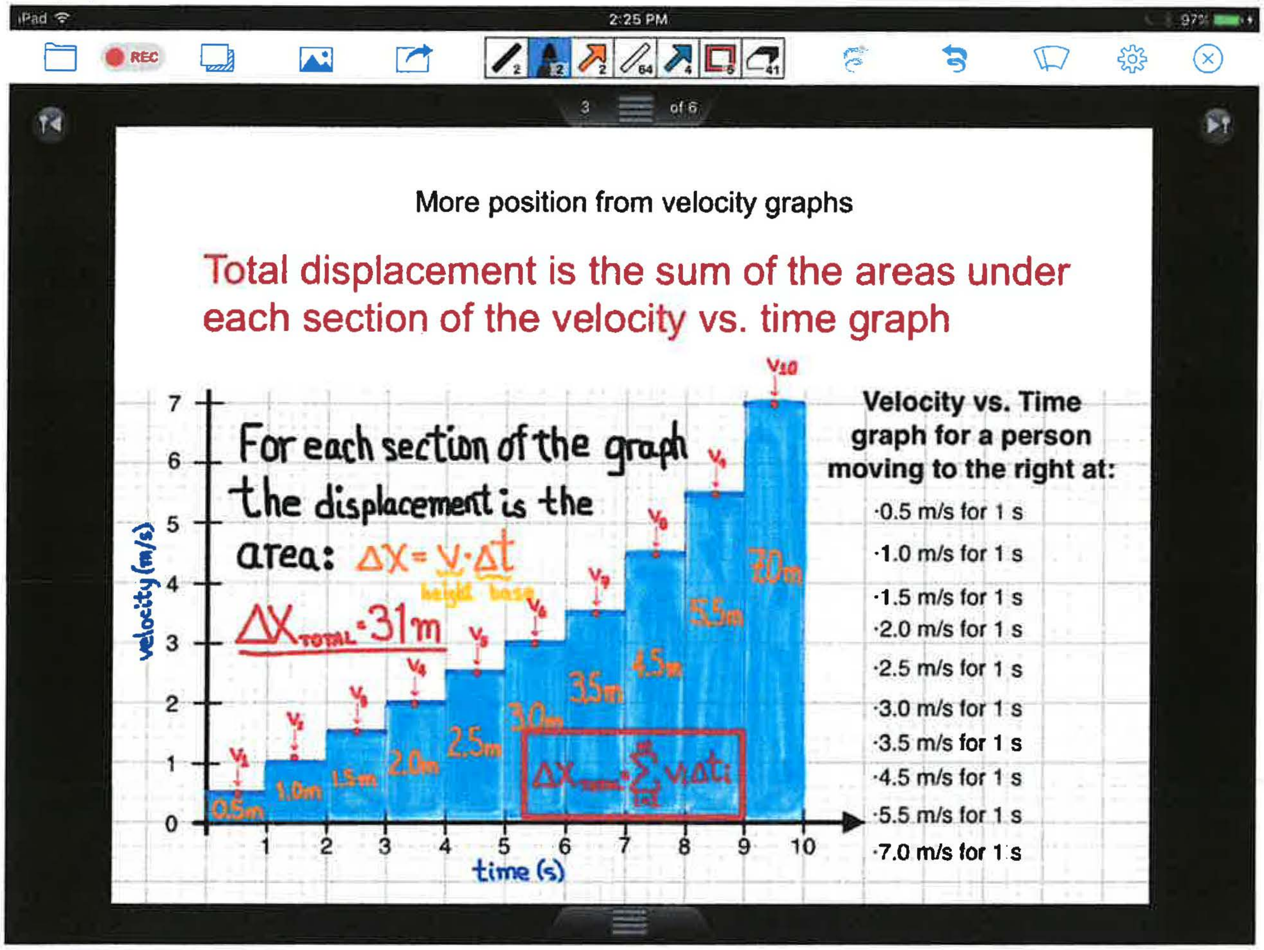


Suppose that our velocity vs. time graph is made up of a bigger variety of velocity steps. How do we find the change in position now?

Consider the graph shown, where someone moves at 0,5 m/s for 1 second, then 1 m/s for 1 s, then 1.5 m/s for 1 second, etc. as shown. How do we find her total displacement?

(TAP to next stop)

Slide 3

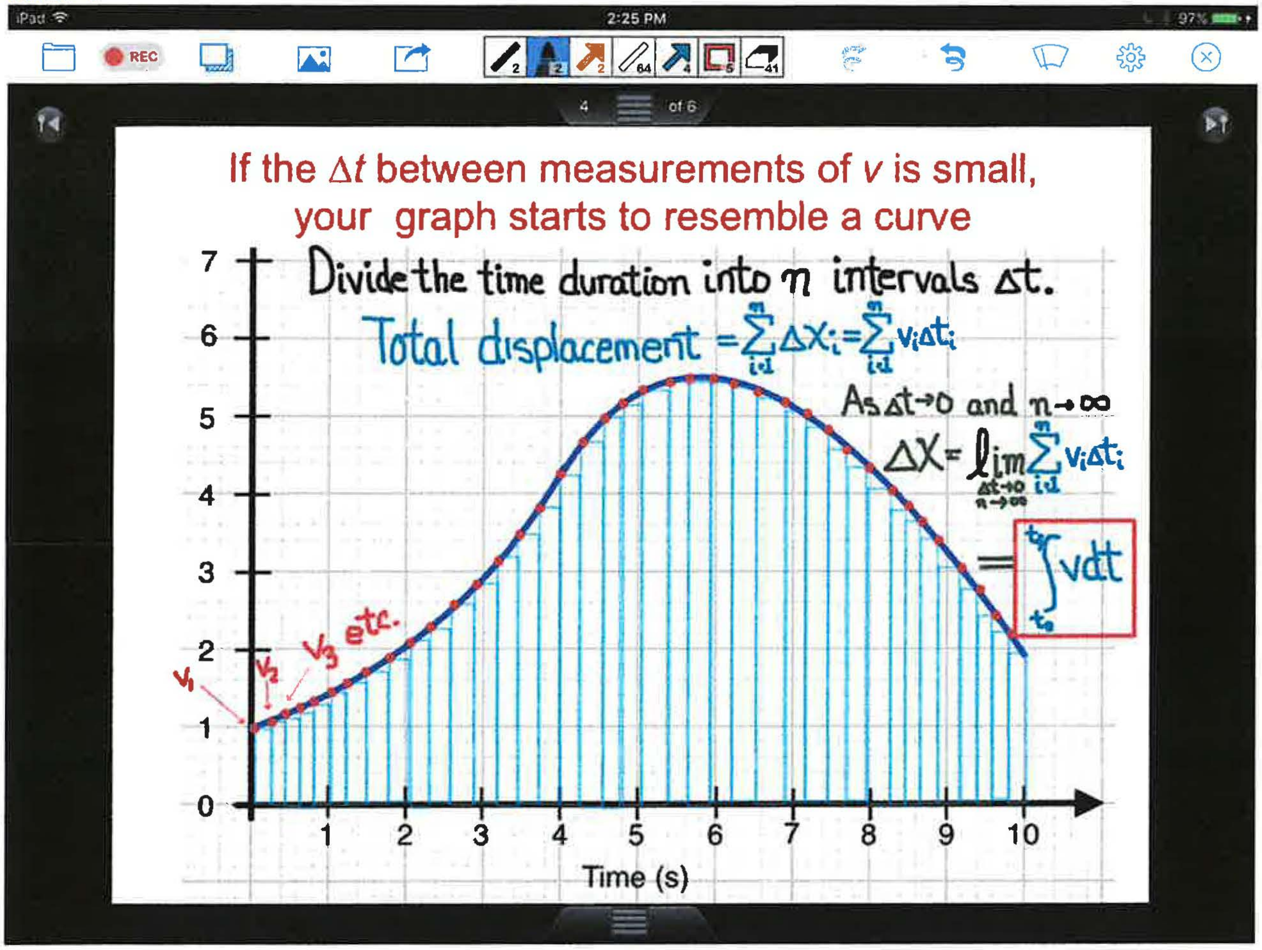


From what we figured out already, the total displacement should just be the sum of the displacements from each of the individual steps, which is just the sum of the areas under each individual section of the graph. **(TAP)**

We can calculate these areas, one at a time, and then add them up.
For the first section, the area is 0.5 m/s high and 1 second wide, for a displacement of 0.5 m.
For the second section , the area is 1.0 m/s high and 1 second wide, for a displacement of 1.0 m.
For the next section , the area is 1.5 m/s high and 1 second wide, for a displacement of 1.5 m and so on.
When we calculate all of the areas and add them up, we get a total of 31 m. Hooray!

(TAP)
This graph is divided into ten different velocity intervals. If we call the velocity of each interval, v1, v2, etc ($v_{i\ \text{in general}}$) up through v10, then we could write the total displacement as the sum of all of these areas **(TAP)** delta x total is the sum over i, for i from one to ten, of the areas v-i times delta t-i.

(disp as integral of v vs. t)
Slide 4



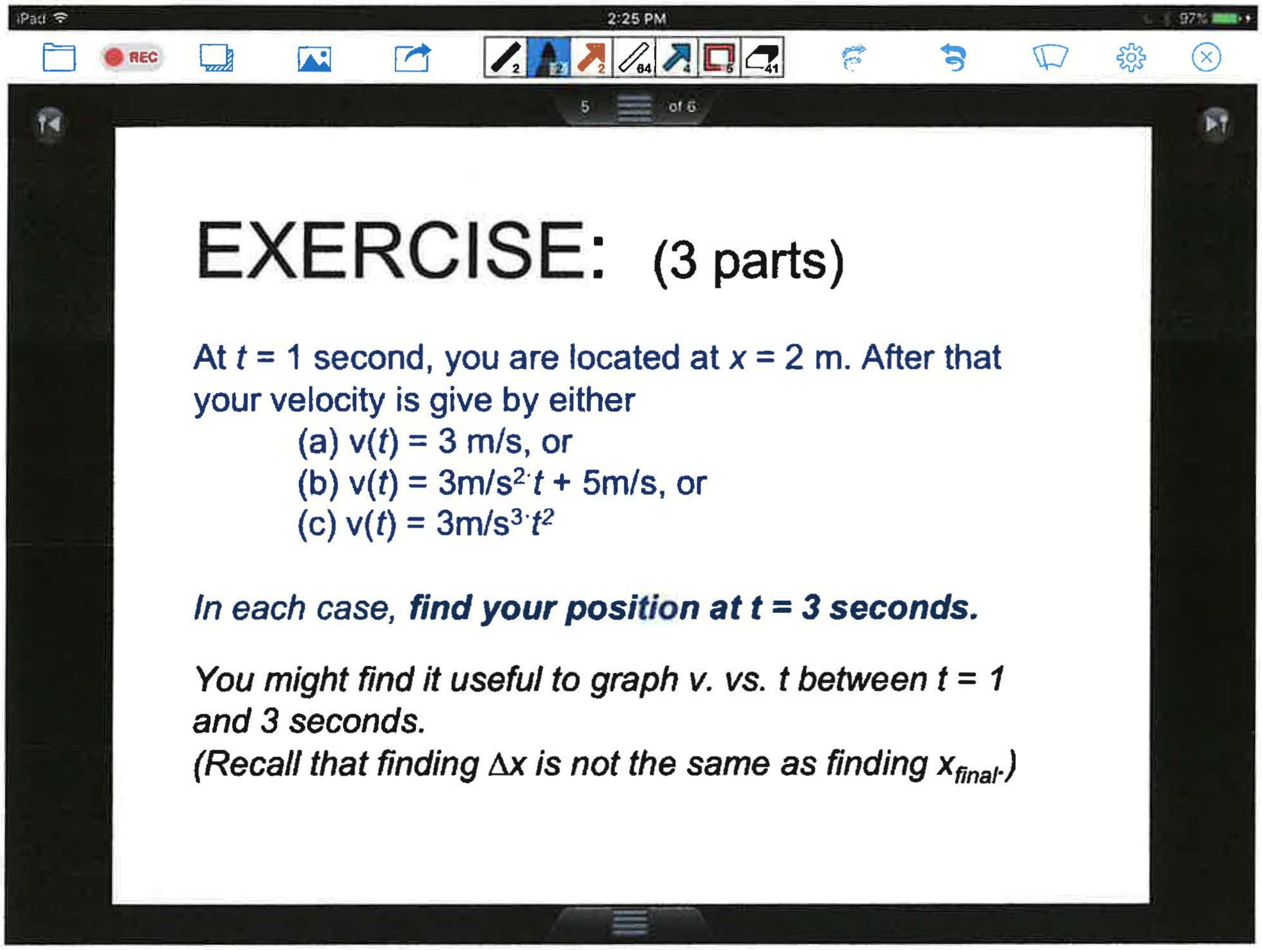
If the time intervals between your measurements of velocity are very small, then your graph starts to resemble a curve.
(TAP)
We could divide the area up into a series of rectangles just like last time, except now the rectangles are much skinnier

Just like before, the total displacement is the sum of the areas of all of these rectangles, each of area v-i times delta t-i.

If we measure velocity over time intervals that go to zero, **(TAP)**
then this sum becomes an integral and the displacement is the area under the *curve*, which is just the integral of v dt.

(exercise delx from v part a.)

Slide 5

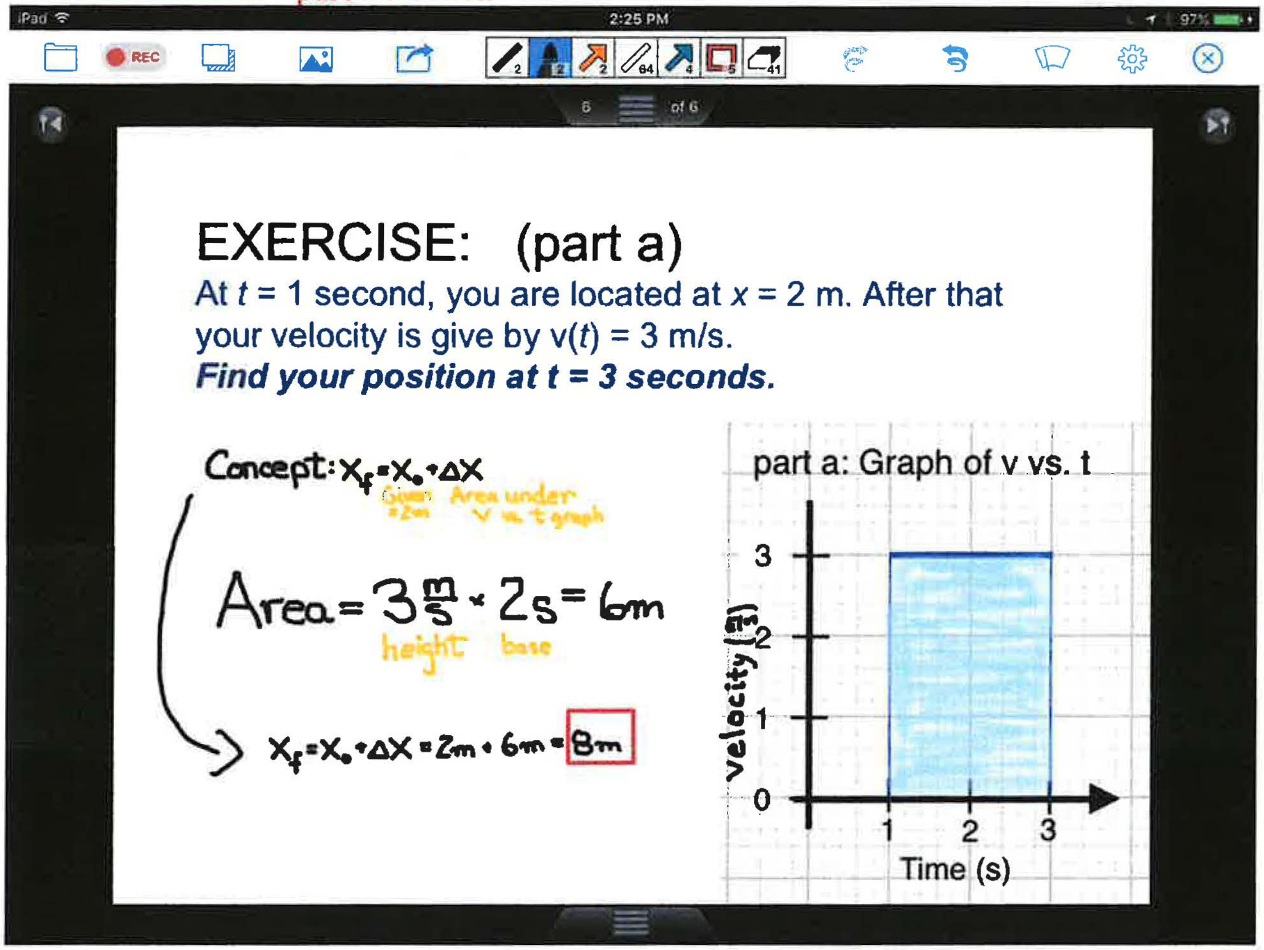


Now you'll try an exercise yourself. In each case the object starts at position x = 2 m. And in each case you are given the velocity as a function of time between t = 1 s and t = 3 s. Your job is to find the position of the object at t = 3 seconds in each of the three cases. I would encourage you to draw the velocity vs time graph and to remember that once you've found delta x you still aren't quite done with the problem.

Try these. You can check in with the solutions when you are done.

Begin new segment

Exercise delx from v part a solution



Greetings! Welcome back!

In part a we are given a starting position x = 2 meters, and a velocity function v of t equals 3 m/s between t =1 and t = 3 seconds, and we want the position at 3 seconds.

(TAP)

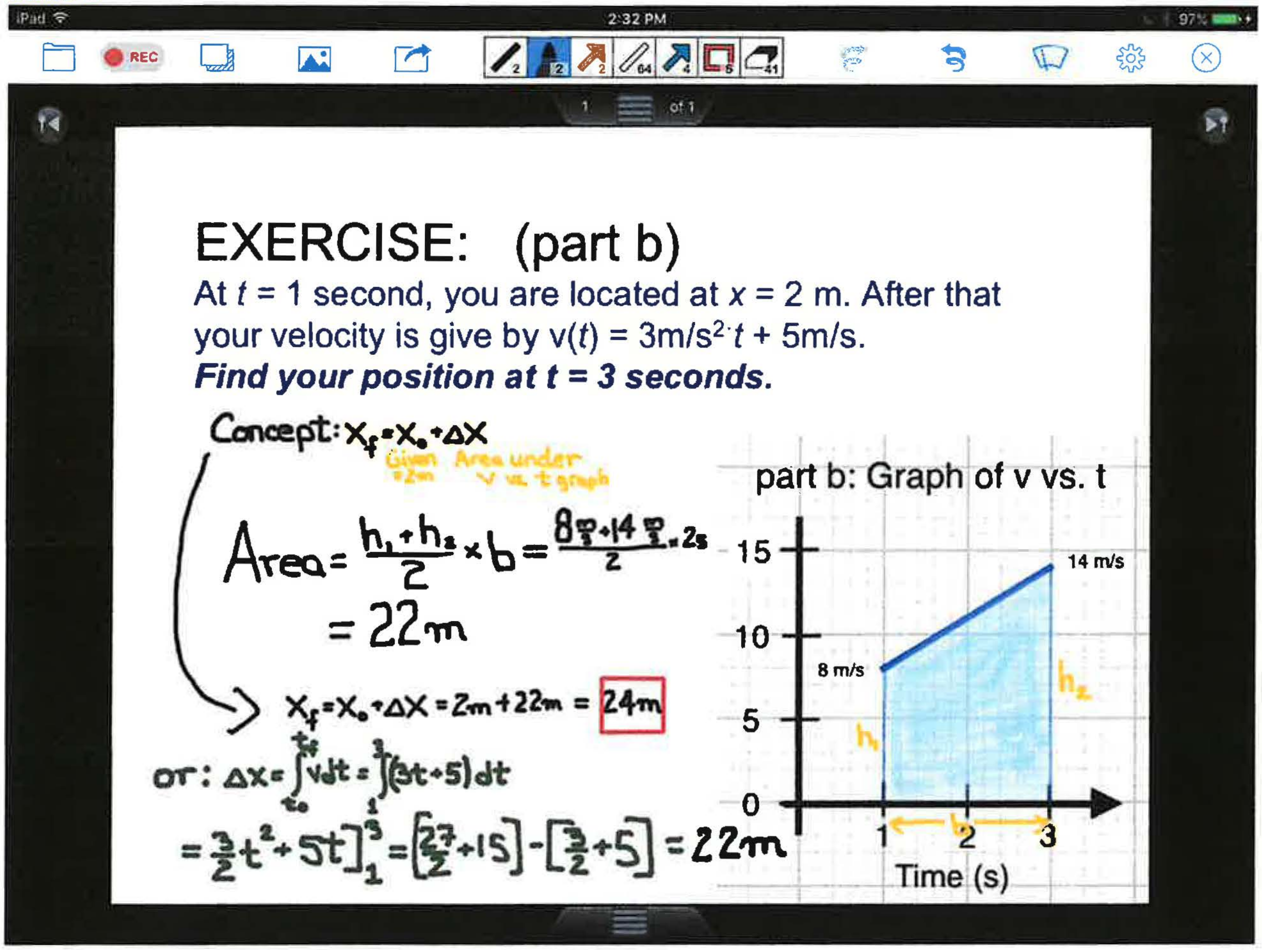
The concept is that we want to find the area under the graph, which will give us the displacement, and then add that to the initial position to get our final position.

(TAP)

The area under the graph is 3 m/s high and 2 seconds wide for a total of 6 meters.
So the final position is x-initial + delta x = 2m + 6 m = 8m. Hooray.

(Exercise partb delx from v)

Slide 2



This time we're given a velocity function v of t equals 3 meters per second squared time t + 5 m/s. The units on the 3 have to be meters per second squared so that when we multiply by t we get something with the right units, m/s.

(TAP)

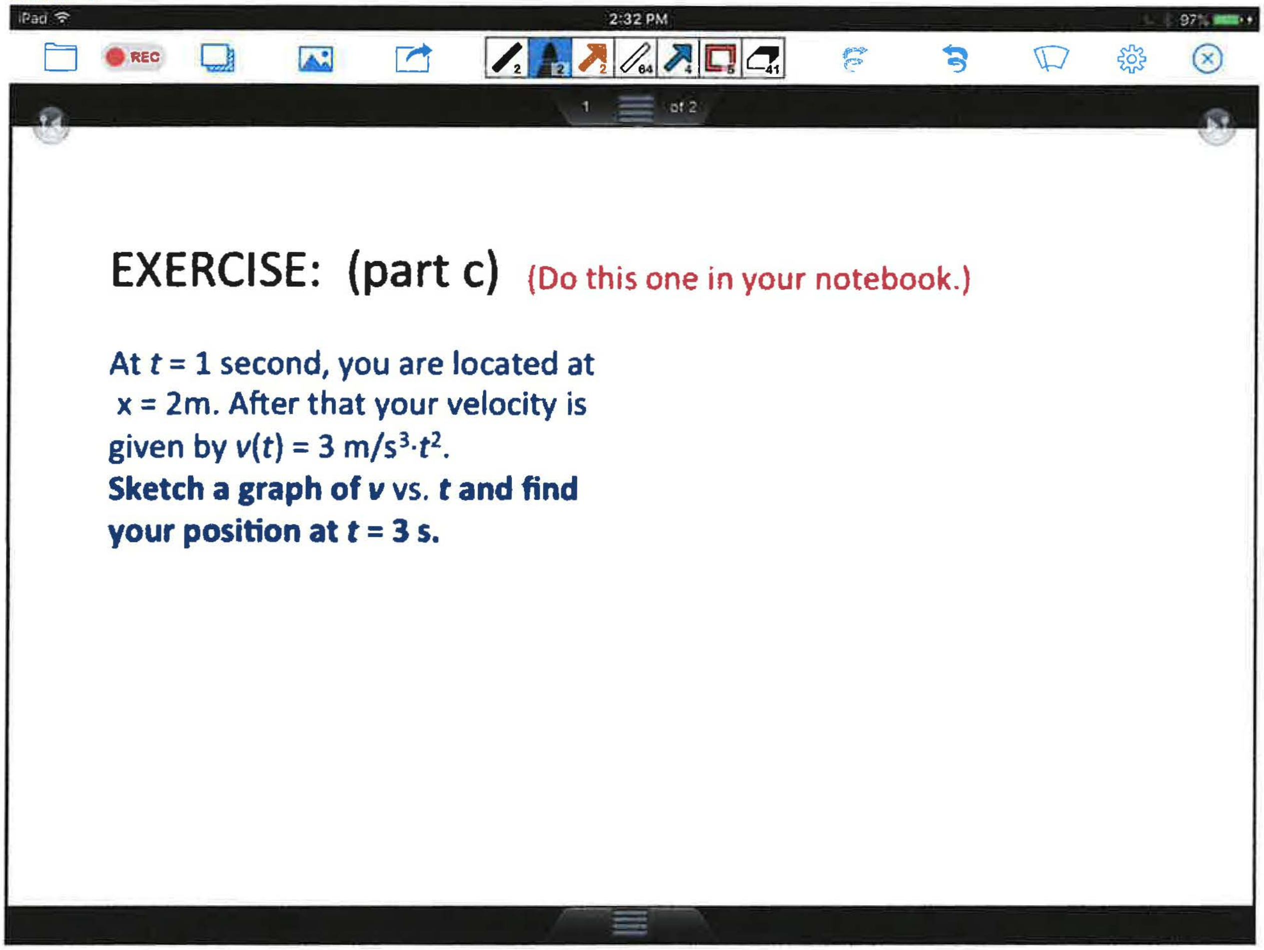
The concept here is the same. Find the displacement delta x and add it to our initial position to get the final position, where delta x is the area under the velocity vs. time graph. In this case our area is a trapezoid. **(TAP)**

The area of a trapezoid is the half the sum of the heights, multiplied by the base. So we have **(TAP)** 8 m/s + 14 ms, divided by 2, time 2 seconds, which gives delta x is 22 m and the final position is 24 m.

(TAP) If we wanted to find the area using calculus, we'd get the change in position by integrating the velocity function between t = 1 and t = 3 seconds, and then add this to the initial position to get our final position. You can confirm that this gives the same result.

(Exercise delx from v part c copy)

Slide 3



Do part c in your notebook. Be sure to include a sketch of the graph of v vs t and to show your calculation of the final position. Do well!

ACCELERATION

Acceleration Lecture videos—Script and Screenshots

The following pages contain the script for the video lectures introducing the topic of acceleration. They were recorded into movies.

The "slides" themselves printed herein represent screenshots from Powerpoint (in which case they could involve animations that won't show up in a static screenshot) or Doceri (in which case the text and images on the screen appear as a sequence of pen strokes or images on the screen). Each screenshot included here is the *final* version of what appears on the screen right at the end of that particular slide. Although these give a sense of what the lecture looks like, best would be to watch the actual lectures themselves. (Links to those are included elsewhere in this report.)

Because I scripted the voiceovers, various words appear in the script to tell me when to advance to the next animation or set of strokes on the screen ("Tap" or Click"); how fast to set the rate at which things appear on the Doceri screen ("Speed 8"); or when to advance through a whole set of strokes all at once in Doceri instead of things appearing one stroke at a time ("Tab to next stop").

In addition, each lecture is a compilation of various slides, some from Powerpoint and some from Doceri, so there are notes at the beginning of some slides as to what presentation or Doceri project to start the narration from.

As a whole, this set of lecture videos comprises half of the first set of videos that students will watch in advance of the second day of class. This script is for ten videos, about 35 minutes total:

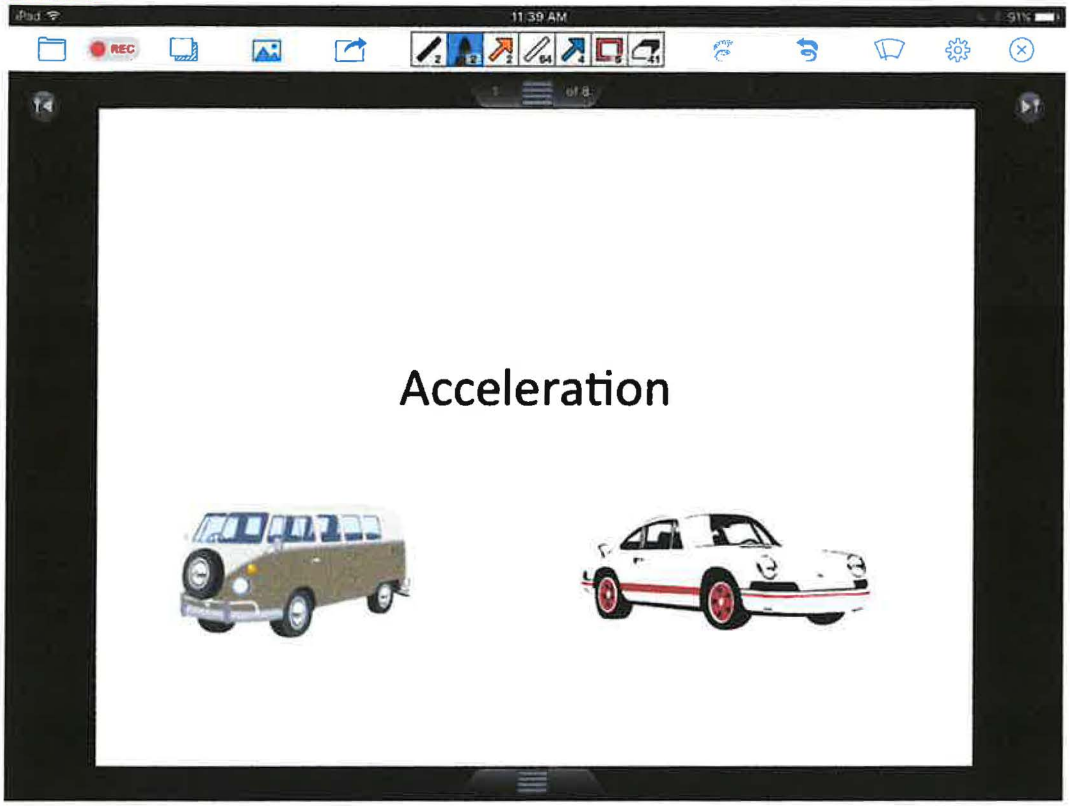
Title	Length (minutes:seconds)
acc intro_1.mov	3:46
acc intro_2.mov	1:26
acc intro_3.mov	2:35
acc intro_4.mov	2:18
acc intro_5.mov	1:52
constant accel_1.mov	5:05
constant accel_2.mov	3:34
constant accel_3.mov	4:06
constant accel_4.mov	5:16
Accel calculus example.mov	4:43

Links for students to access the videos are posted in Moodlerooms. I also post a student version of the script, which is essentially all of the text in what follows but without any of the "Tab" or "Click" instruction. The student version is separated out by lecture movie title rather than by the name of the source file.

Acceleration Script

Acc intro

Slide 1



Greetings!

(TAP)

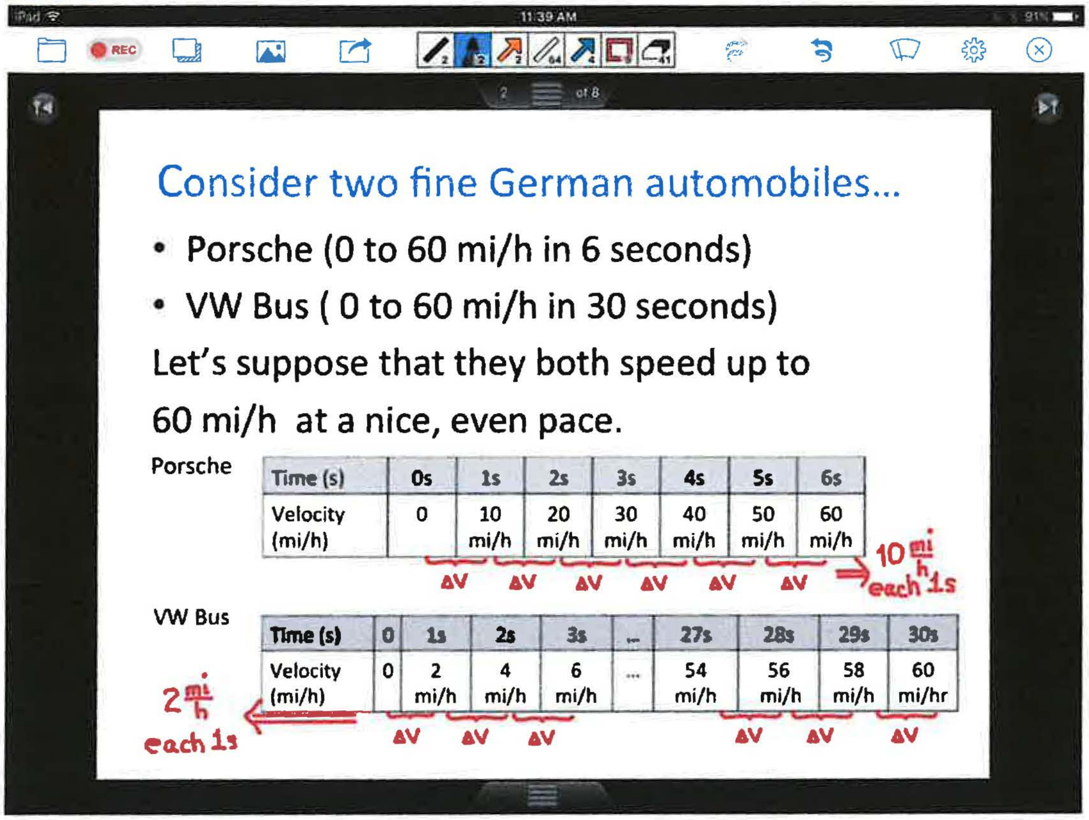
We'll start off our exploration of acceleration by imagining two fine German automobiles that are both speeding up from rest at a nice, steady pace.

The Porsche goes from 0 to 60 miles an hour in 6 seconds, while the VW bus needs 30 seconds to go from 0 to 60 miles per hour.

So we can imagine that their speed increases like this.

(TAP)

Slide 2



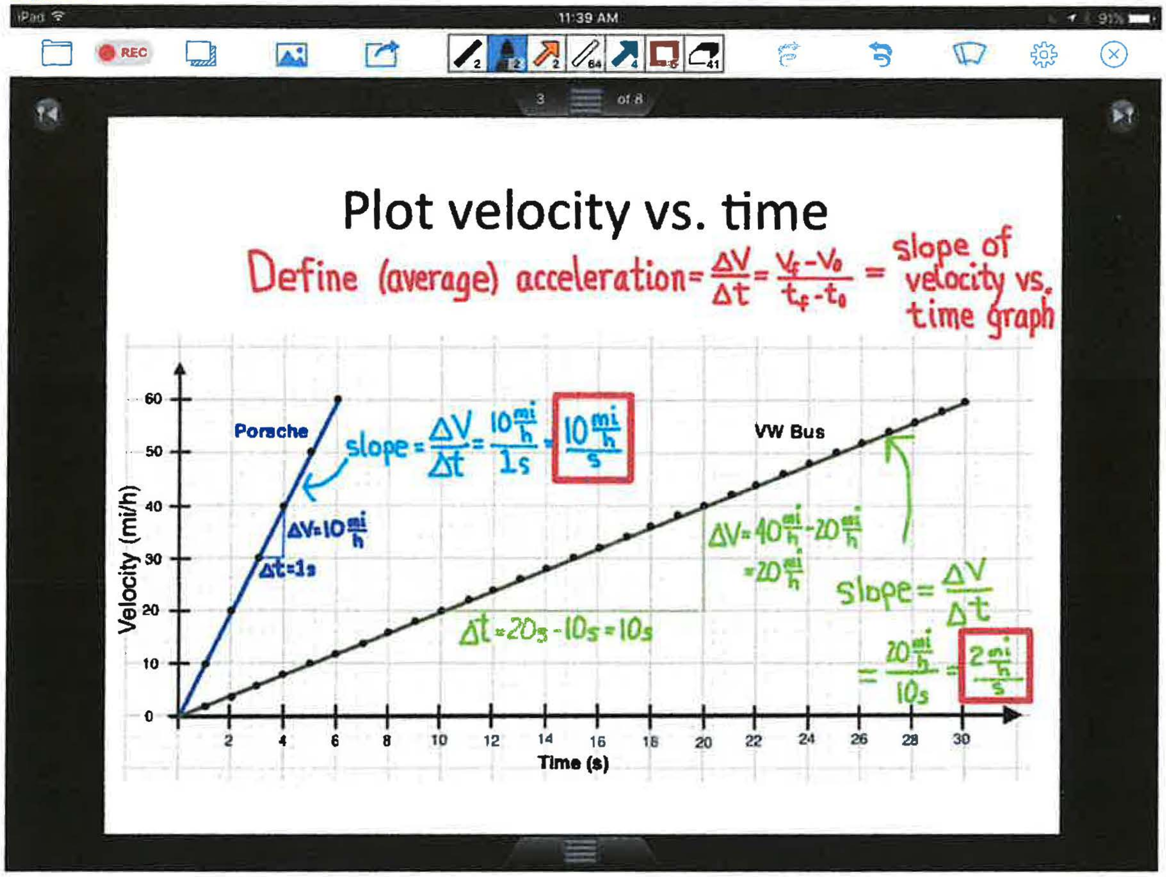
After 1 second the Porsche is going 10 mi/hr, after two it is going 20 miles per hour, and so on up to 60 miles per hour in 6 seconds.

The bus . . . well, after one second the bus is racing along at 2 mi/hr, at 2 seconds it is going 4 mi/hr, etc. until it reach 60 mi/hr after 30 seconds.

Clearly the Porsche gets to 60 mi/hr sooner! How do we describe how quickly these two cars speed up?
One way is to look at the change in velocity of each car during each second.
(TAP)
For the Porsche, the velocity is changing by 10 mi/hr each second.
(TAP)
For the Bus, the velocity is changing by 2 mi/hr each second.

(TAP)

Slide 3



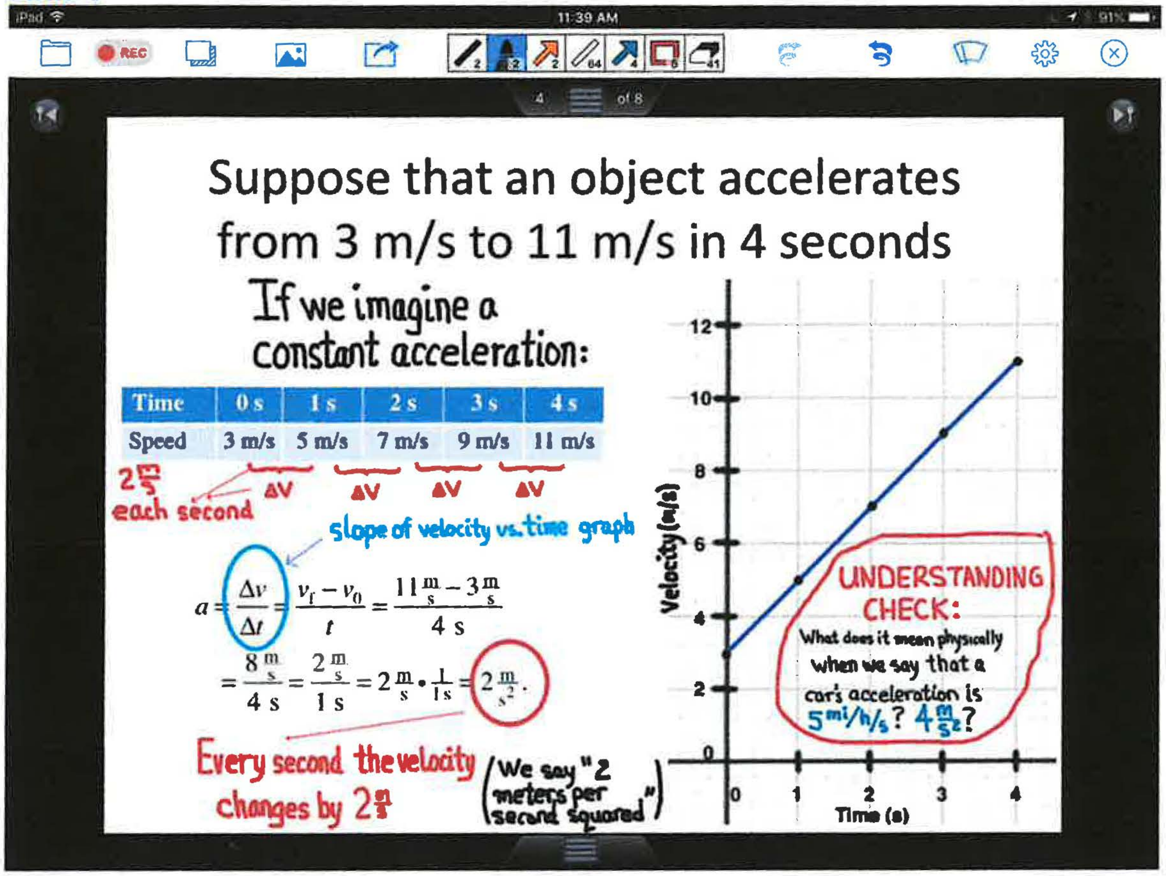
Here are velocity vs. time plots for the two cars. Notice that the line representing the speed for the Porsche is much steeper than the one for the Bus. Let's calculate the slope for these two lines.
(TAP)
For the Porsche we get 10 mi/h/sec and for the bus we get 2 mi/h/sec, exactly what we got when we looked at the change in velocity each second in the data table.

Our common-sense notion of acceleration is that however we define it, it should be a bigger number for the Porsche than for the bus.

(TAP)
It makes sense to define acceleration as the change in velocity with time, delta v over delta t, which is the slope of the velocity vs. time graph.

This is really a definition of *average* acceleration, because in general the velocity between two points separated by a time delta t will not be changing steadily.

(Acc intro part2)
Slide 4



Let's consider another object increasing velocity steadily from 3 m/s to 11 m/s in four seconds.

Tap

We can imagine that the velocity increase goes like this: 3 m/s at the start, 5 m/s one second later, 7 m/s one second after that, and so on.

Tap

If we look at the change in velocity each second, we see that it is 2 m/s each second.

Tap

We can also graph the data and take the slope of the graph. We get 2 m/s per second, or 2 meters per second squared.

I'm not suggesting that there is any such thing as a square second! It's just that when you divide meters per second by seconds you gets seconds time seconds on the bottom, which is seconds squared.

Tap

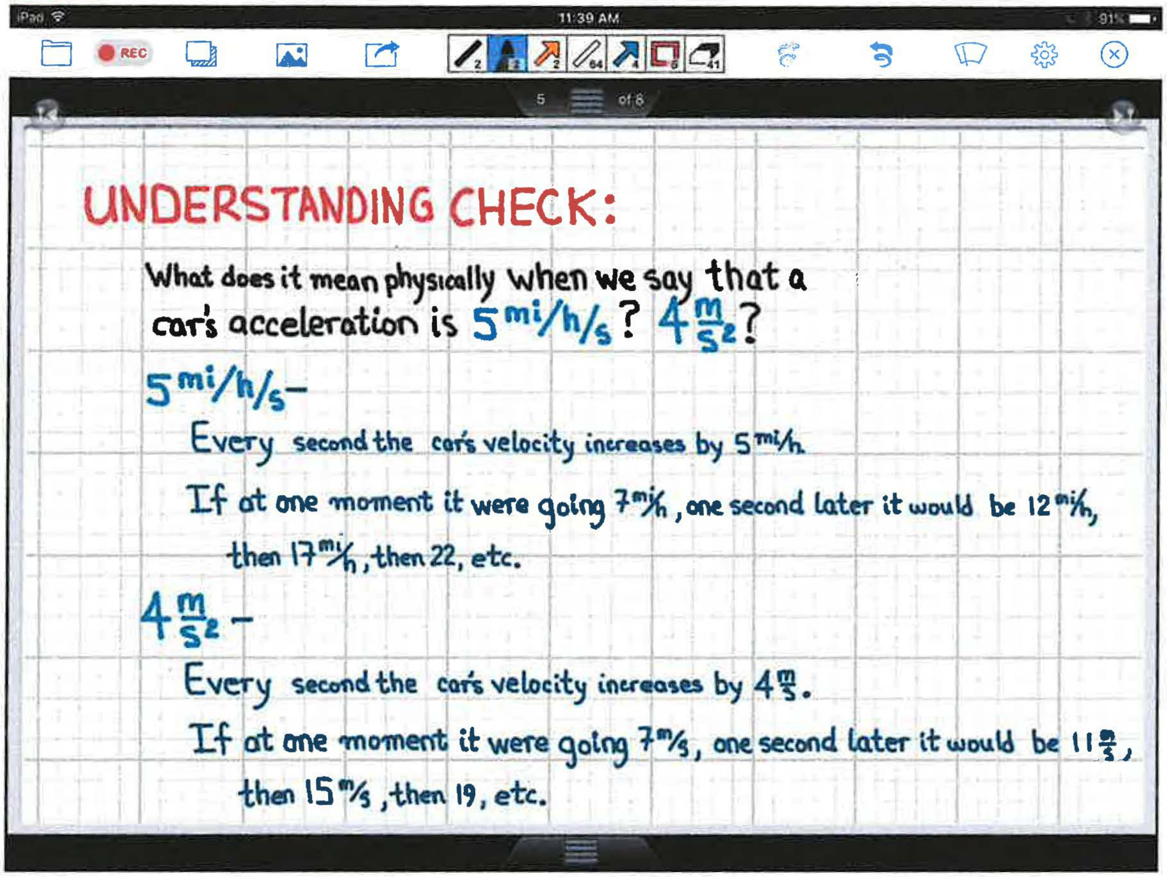
What does it mean physically when we say that a car's acceleration is 5 mi/h/s? What about 4 meters per second squared?

"Physically" here means that your answer should tell us something about how the car's velocity is changing. Saying "It's acceleration is 5 mi/h/s" doesn't communicate that you know what 5 mi/hr/second means.

Give it a shot. I'll see you in the next segment.

Begin new segment

Slide 5



Tap to first stop.

Greetings! Welcome back.

We were looking at the question of what it means physically to say that a car's acceleration is 5 miles per hour per second, or 4 meters per second squared.

Tap

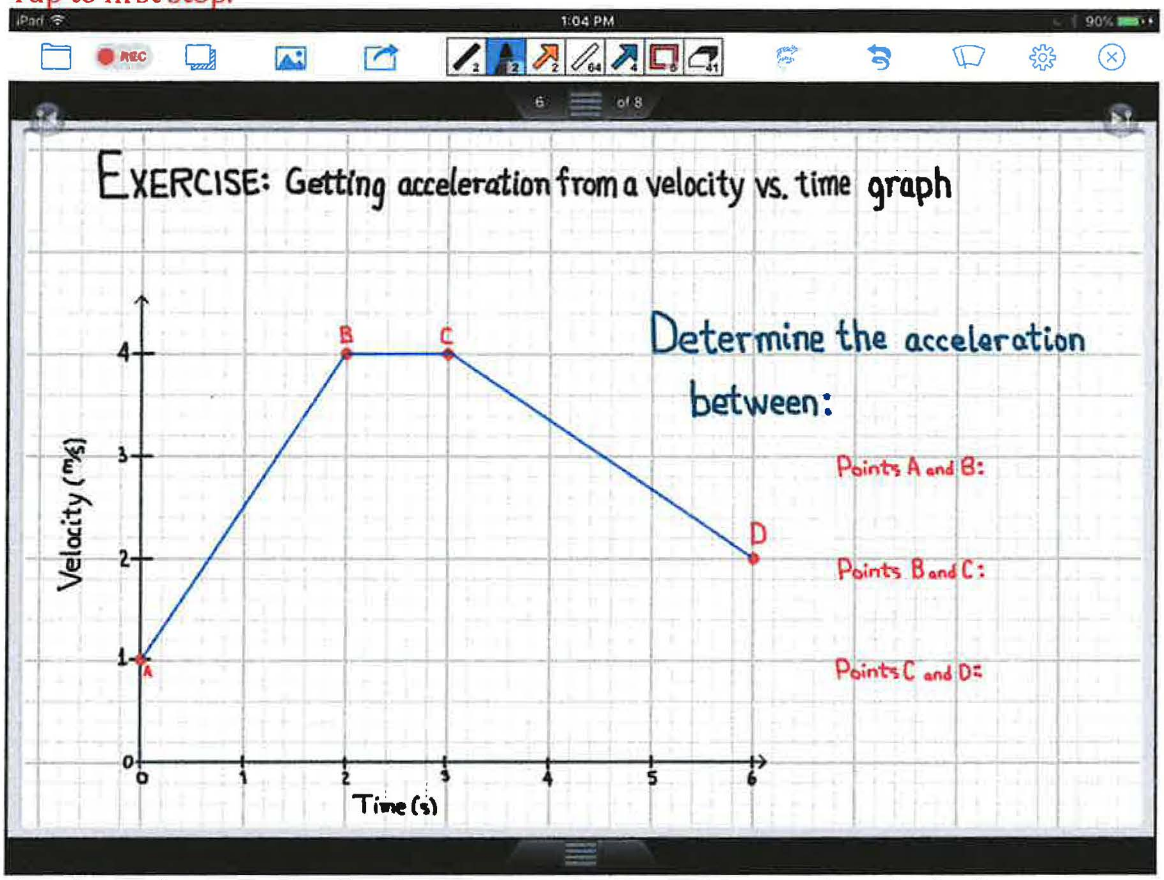
Physically, 5 miles per hour per second means that every second the velocity of the car increases by 5 miles an hour. If at some moment it were going 7 miles an hour, one second later it would be going 12 miles an hour, and one second after that, 17 miles and hour, and then 22 and so on.

Tap

Physically, 4 meters per second squared means that every second the velocity of the car increases by 4 meters per second. If at some moment it were going 7 meters per second, one second later it would be going 11 meters per second, and one second after that, 15 meters per second, and then 19 and so on. Ever second the velocity is 4 meters per second greater than it was the second before.

Slide 6

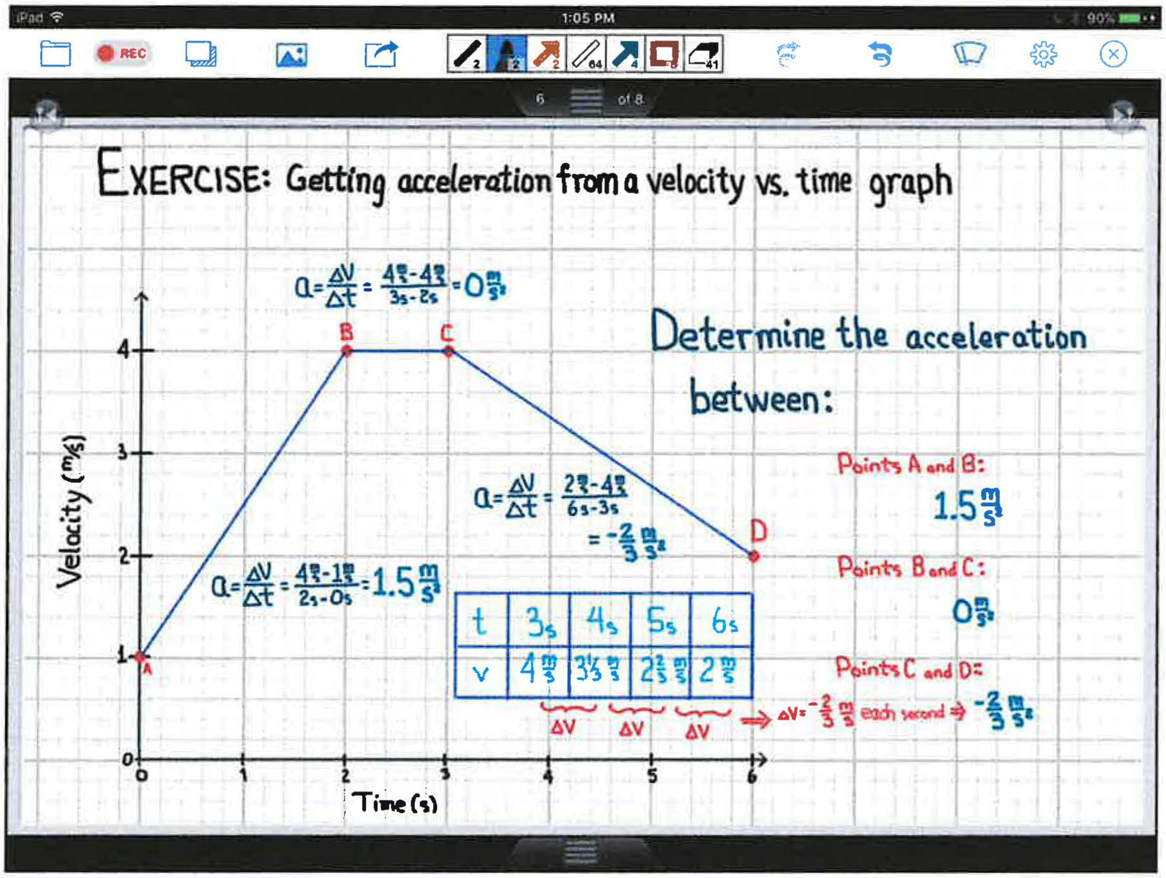
Tap to first stop.



Here's another exercise. You're given a velocity vs time graph and asked to determine the acceleration for each segment of the graph—between points A and B, B and C , and C and D.

GO for it! I'll see you in the next video.

Same slide. Start the movie where we left off.

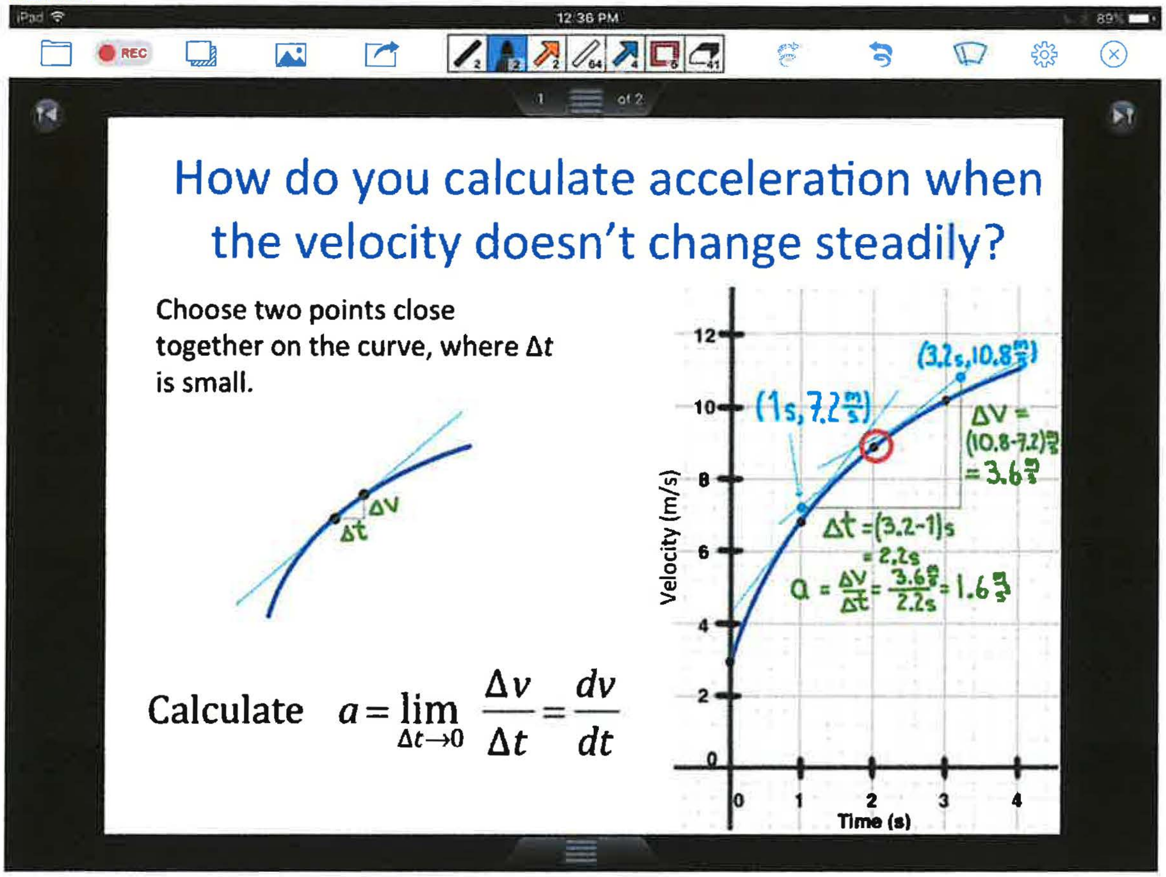


Greetings! Welcome back.
You were given a velocity vs time graph and asked to determine the acceleration for the three different segments of the graph.

The concept involved is the definition of acceleration as the change in velocity with time, which is the slope of the velocity vs. time graph. So all we need to do is calculate the slope of each section of the graph.
(TAP)
From A to B, the change in velocity with time is vfinal minus v-initial divided by t final minus t-initial, which is 4 m/s minus 1 m/s over 2 s minus 0 s which gives 1.5 m/s^2.
(TAP)
From B to C, delta v over delta t is 4 m/s minus 4 m/s over 3 s minus 2 s which gives 0 m/s^2. This means that the velocity isn't changing.
(TAP)
And from C to D, delta v over delta t is 2 m/s minus 4 m/s over 6 s minus 3 s which gives negative two-thirds m/s^2.
If you look at the last one, it tells us that each second the velocity changes by negative two-thirds meters per second.

(Acc intro part3)

Slide 7



Now we'll look at how to calculate acceleration when the change in velocity isn't constant

We're looking at a situation where the slope of the velocity vs. time graph isn't constant. This is the same situation we dealt with for position graphs when the velocity (which is the slope of the position vs. time graph) wasn't constant.

(TAP)

Our approach then was to choose two points close together, where delta t was small, and then calculate the acceleration as the limit of the value of the slope as delta t approaches zero (that is, as these two points become infinitesimally close together.

(TAP)

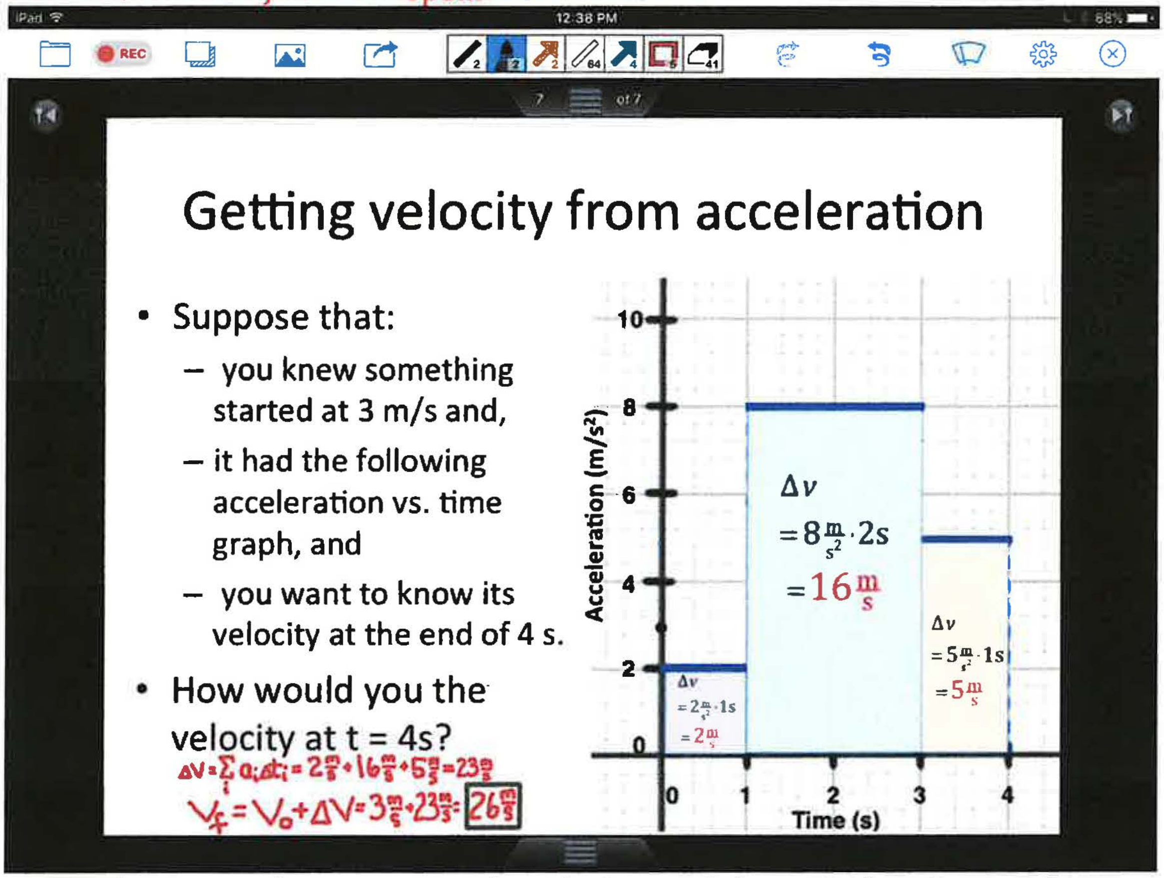
For the three points shown on the graph, we can draw tangent lines to the velocity curve at each point. The acceleration at each of those moments in time is the slope of the tangent line at that point.

For example, if we wanted to know the acceleration at t = 2 seconds, we could draw the tangent line (which we already did) and then calculate the slope by choosing two points on that tangent line.

(TAP)

Here I've done that, and you can see that the slope turns out to be 1.6 m/s^2.

Acceleration Intro June Powerpoint slide number 6



Now we're going to look at getting the velocity graph from acceleration graphs. Here we're told that at t = 0 seconds something is going 3 m/s, and its acceleration is given by the graph, and we want to know the velocity at the end of 4 seconds.

We could approach this by looking at each part of the graph individually.
For the first part of the graph you accelerate at 2m/s each second, and we're going to do that for only one second, so our change in velocity is 2 m/s.
Click
This also happens to be the area under the acceleration vs. time graph.

For the next section of the graph you accelerate at 8m/s each second, and we're going to do that for two seconds, so our change in velocity is 16 m/s.
Click
This also happens to be the area under that part of the acceleration vs. time graph.

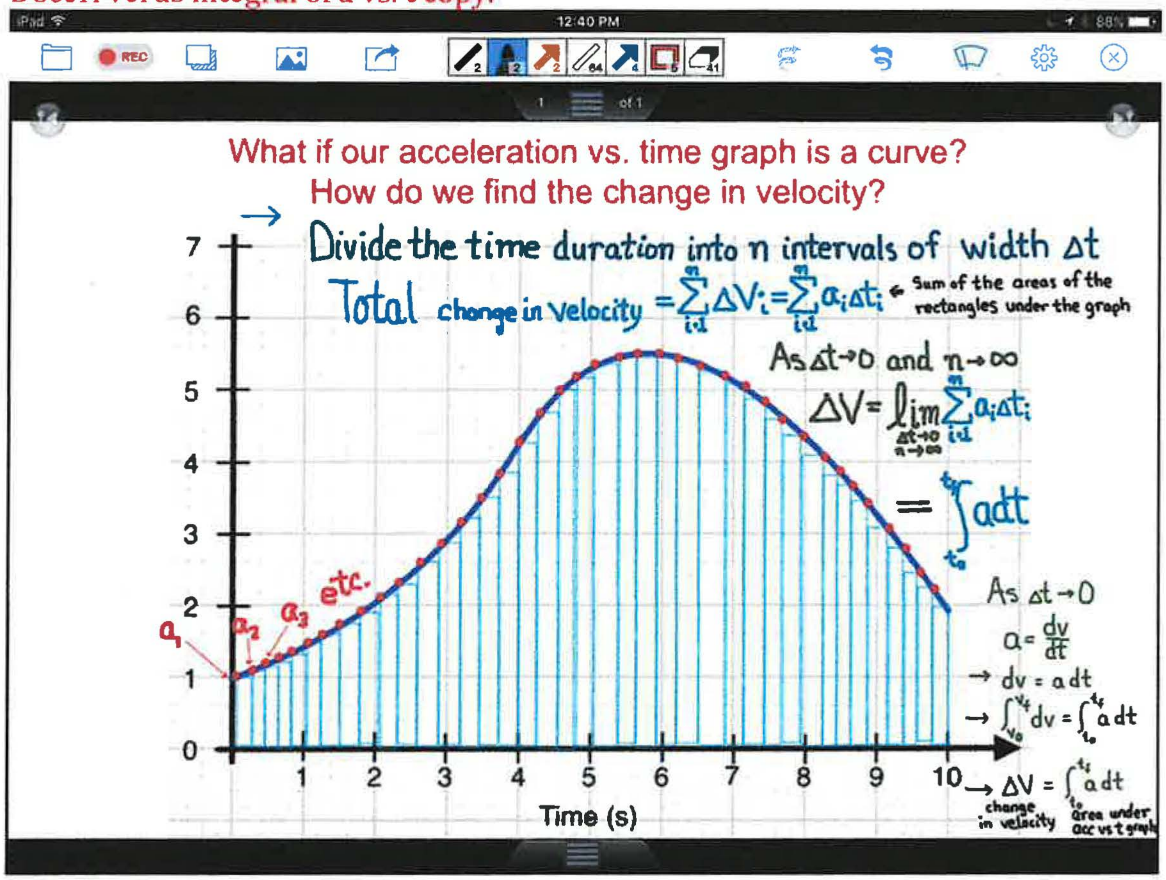
And for the last section of the graph you accelerate at 5m/s each second, and we're going to do that for only one second, so our change in velocity is 5 m/s.
Click
This also happens to be the area under that part of the acceleration vs. time graph.

The total change in velocity is the sum of the individual changes in velocity, which is the same as the sum of the areas under each section of the graph.

Click

So the total change in velocity is 2 m/s + 16 m/s + 5 m/s = 23 m/s and the final velocity is the initial velocity plus the change in velocity, 3 m/2 plus 23 m/s, or 26 m/s all together.

Doceri vel as integral of a vs. t copy.



Click (makes graph show up)

How do we find the change in velocity if the acceleration graph is a curve?

We'll use the same approach as we did in the last slide. The change in velocity is the area under the acceleration vs. time graph.

Click to next stop

We break the time duration of the motion into a large number of intervals, each of width delta *t*.

Click

Then we'll draw in rectangles, each with height a sub i and width delta t

The area of the first rectangle is a1 times delta t, the area of the second rectangle is a2 times delta t.

In general, the *i*-th interval will have height a_i and width delta t, so the change in velocity for each interval is a sub i times delta t.

Click

The total change in velocity will be the sum of the areas of each interval.

Click

As delta t goes to zero and the number of intervals goes to infinity, this sum of the area of the rectangles becomes the integral of a dt.
This is calculus!
Speed 2, Click
As delta t goes to zero, a becomes dv dt.
Multiply both sides by dt, then integrate dv from v-initial to v-final, and a dt from t initial to t final.
The first integral gives you delta v, and the second one gives you the area under the acceleration vs. time graph. Hooray!

Acceleration Intro Powerpoint slide number 8
Here is an exercise to do in your notebook.
You are given an initial velocity and a graph of the acceleration as a function of time, and asked to find the final velocity.
Go for it.

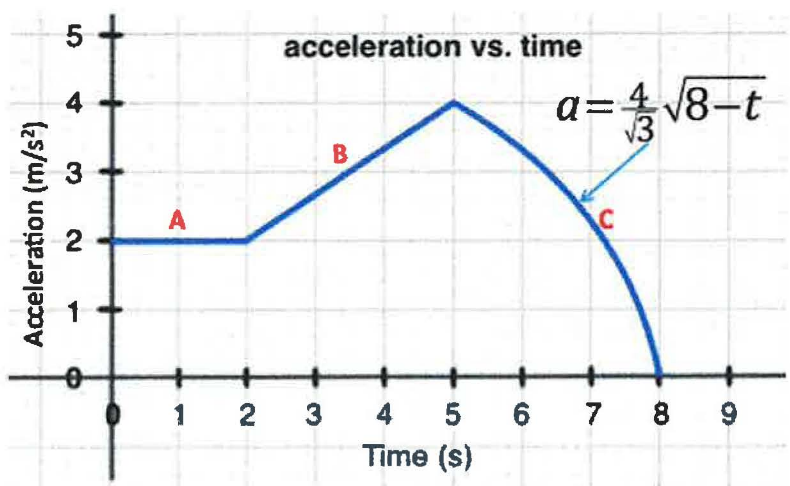
EXERCISE (Do this one in your notebook.)

The graph show the acceleration vs. time behavior of an object initially moving in the +x direction.

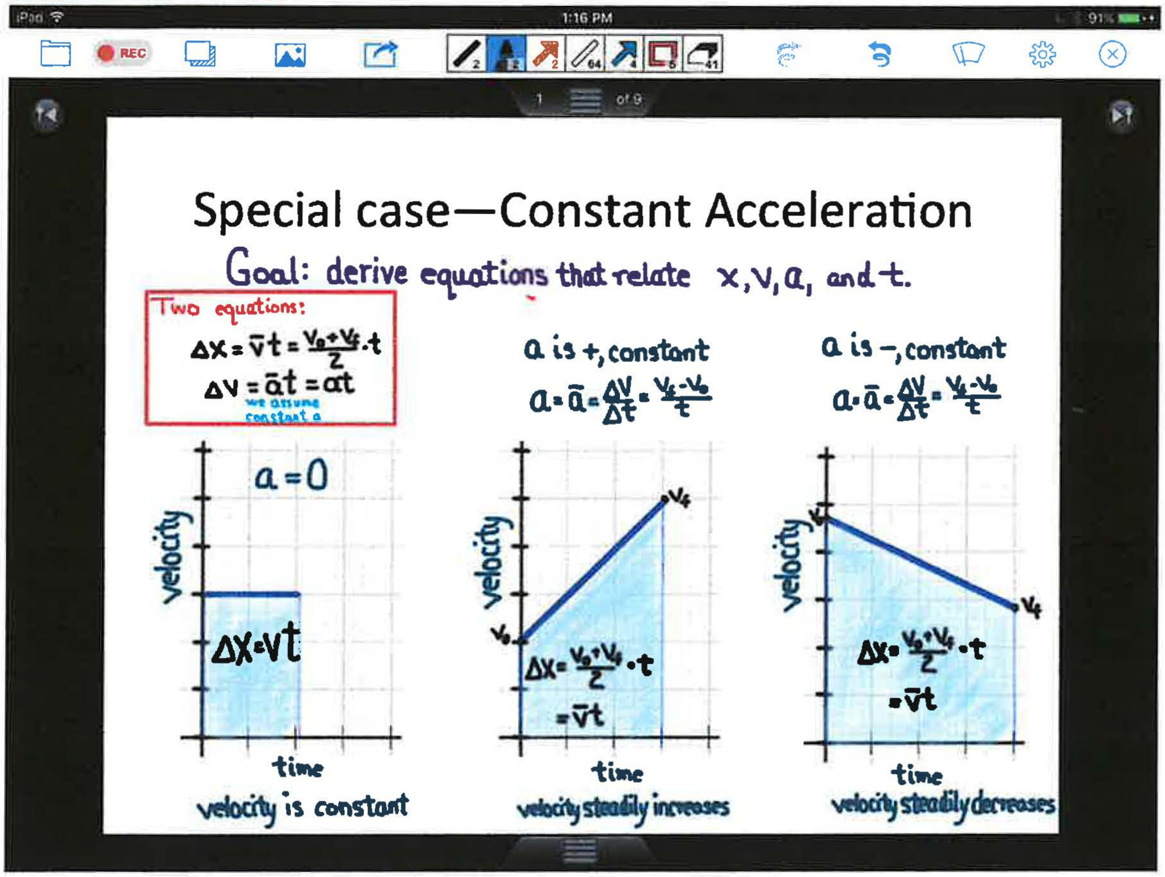
For each section of the graph (A, B, and C):

- **Calculate the change in velocity of the object**

Calculate the final velocity if the initial velocity was +12 m/s.



Begin new segment
constant acc 1 Doceri



We're going to consider a special case—situations in which the acceleration is constant.
TAP to next stop
Our goal is to derive algebraic equations that we can use for that special situation of constant acceleration.

We'll consider three cases:
TAP to next stop
One in which the velocity is constant (so the acceleration is zero)
, one in which the velocity increases at a steady rate (a is positive)
, and one in which the velocity decreases at a steady rate (a is negative)

TAP
In the constant velocity case, the area under the graph is just v times t, so we could write delta x equals v t
TAP
The area under the increasing velocity graph is a trapezoid. Its area is the average of the two heights times the base, so here delta x is v-zero + v-final over 2, times t.
In the constant velocity case, the area under the graph is just v times t, so we could write delta x equals v t

TAP

The area under the decreasing velocity graph is also a trapezoid so the formula for its area is the same.

TAP

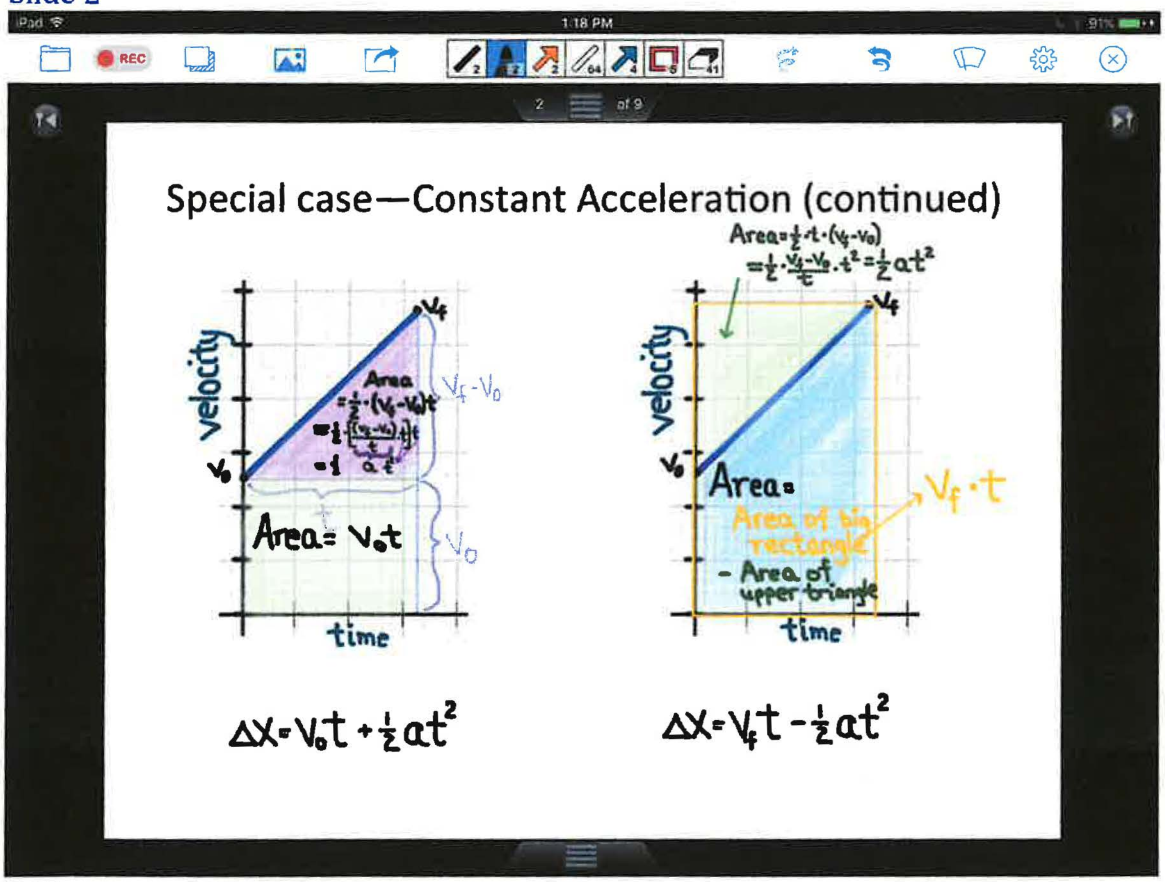
So so far we have derived two equations for motion with constant acceleration:

The first is delta x equals average speed times time

The second is delta v equals average acceleration times time. Since we're assuming the acceleration is constant we can drop the average sign.

(constant acc2 Doceri)

Slide 2



TAP

There are two more equations we can derive for constant acceleration based on the area under a velocity vs. time graph.

In each case we'll consider a case where the velocity is increasing.

For the first case

TAP

We break the area into a rectangle of height v-zero and base t, and a triangle of height v-final minus v initial and base t.

TAP

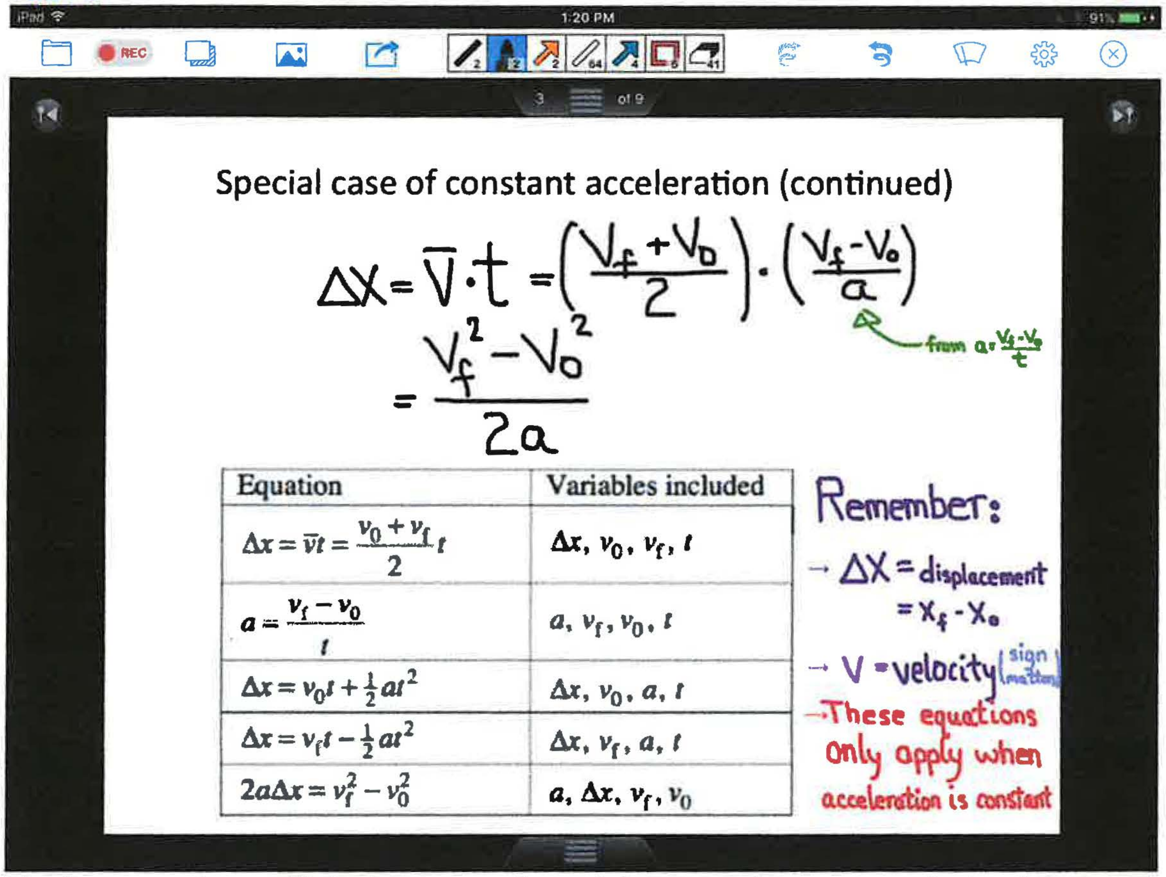
The area of the lower rectangle is v-zero t.

TAP
The area of the triangle is one-half times v-final minus v zero times t.
TAP
We can divide and multiple the v-final minus v zero term by t.
TAP
v-final minus v zero divided by t is the acceleration a, and there is a t-squared now.
TAP
So the area of the triangle is one half a t squared, and the total displacement delta x is given by v-zero t plus one half a t squared.

TAP
In the second case, the area under the graph is the difference between the area of the big yellow rectangle *minus* the area of the upper triangle.
TAP
The area of the large rectangle if v-final times t.
TAP
The area of the upper triangle is one-half a t-squared, just like before.

TAP
So the delta x is the total area under the velocity vs time graph, which is v-final t minus one half a t squared.

(constant acc3 Doceri)
Slide 3



The last equation we'll derives starts from delta x equals average velocity times time.
We substitute v-final plus v-zero over 2 for the average velocity, and v final *minus* v-zero over a for the time.
The top is a difference of two squares, so we get
TAP
V final squared minus v-zero squared divided by 2 A

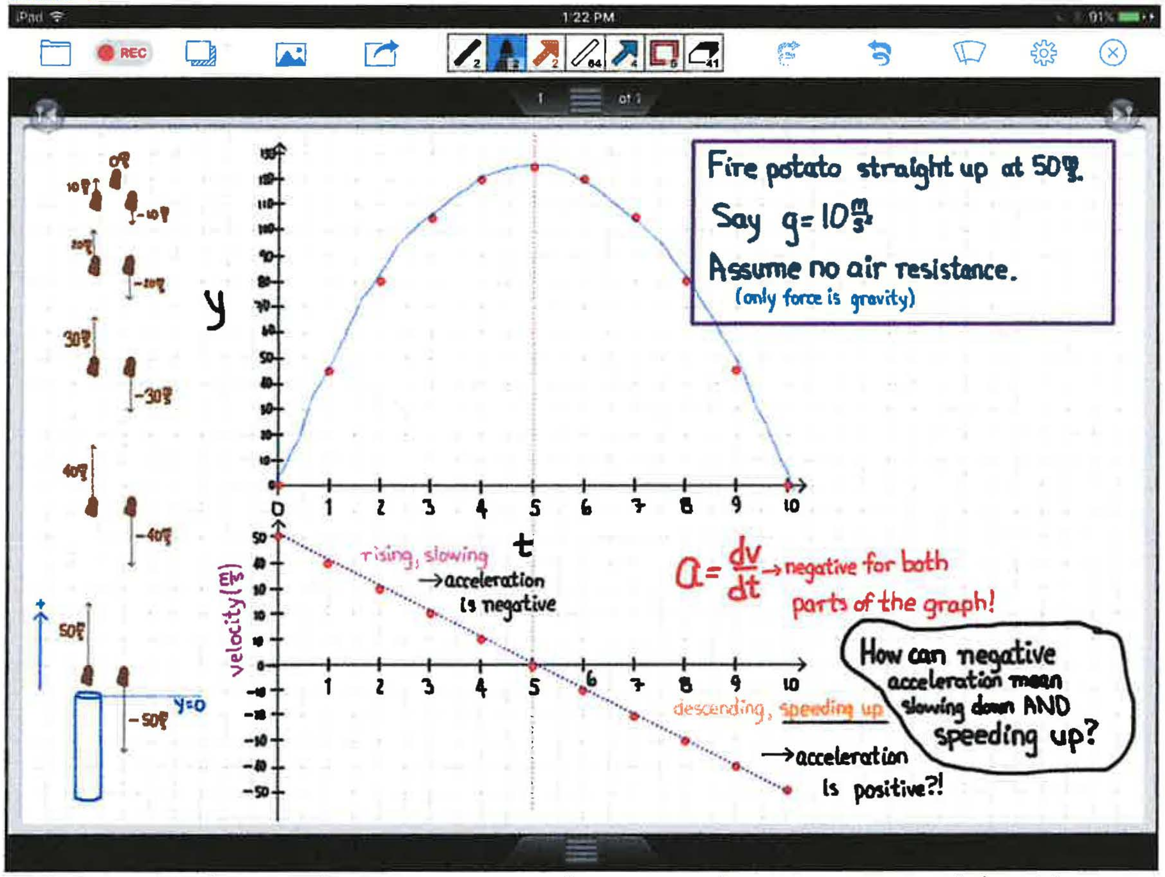
We have developed five equations for the special case of constant acceleration. Here there are:
TAP
Notice that there are five variables total: delta x, V-initial, v-final, acceleration, and time.
Each equation contains four of the variables.
So in any problem you are given three of the variables and need to solve for the fourth one.
It really isn't physics at this point. It comes down to picking the right equation, rearranging for what you want, and plugging in the numbers.

There are a couple of important things to remember:

Delta x is a displacement, not a distance. If you end up where you start, delta x is zero
Next
TAP
V is a velocity, not a speed. Once you decide what direction is positive, you'll need to attach the correct sign to v before you plug anything in.
TAP
And finally, these equations only apply when the acceleration is constant. This was always the case in Physics 2AG, because you didn't have any calculus yet. Not that we have more powerful tools we can solve more interesting problems.
Again, these aren't fundamental equations of Physics. They are equations that apply in special case where the acceleration is constant.

(Intro negative acc Doceri)

Slide 4



Let's imagine that you fire a potato straight up out of a potato gun, and that it's initial speed is 50 m/s.

And let's imagine that the acceleration due to gravity is 10 m/s².

And let's imagine that there is no air resistance, so that the only force acting on the potato once it leaves the gun is gravity.

TAP

On the way up the speed decreases by 10 m/s each second, and on the way down the speed increases by 10 m/s each second.

TAP

We can choose the top of the potato gun to be our origin (y = 0) and upward to be the positive direction, and then we can plot the position and velocity of the potato as a function of time.

TAP

What's clear is that on the way up the potato is slowing down, and its acceleration is negative. This makes sense if we think of negative acceleration as "slowing down".

TAP

On the way down, the potato is speeding up, going faster. Does this mean that the acceleration is positive? This would make sense if positive acceleration meant "speeding up".

TAP

If we look at our definition of acceleration as the slope of the velocity vs. time graph, we can see that the slope of the graph is *negative* for both the rising and falling parts of the potato's motion.

So the question is

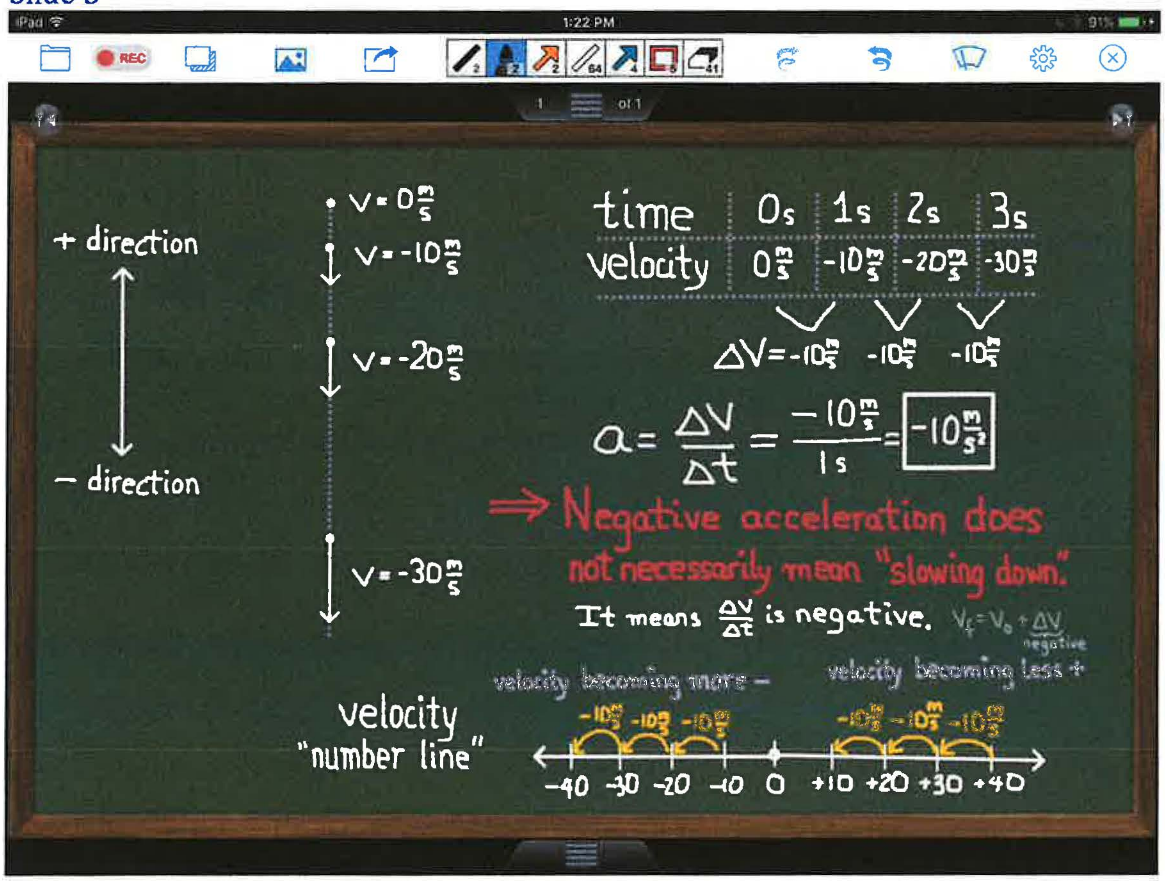
TAP

How can negative acceleration mean both slowing down and speeding up? What does "negative acceleration" mean?

Let's look at this more closely.

(explanation neg acc Doceri)

Slide 5



One of the first things we do in a problem is set up a coordinate axis—we decide what direction is going to be the positive direction. For the potato we decided upward would positive.

Speed 4 TAP

When the potato had reached the top of its flight and was descending, its velocity went 0, then minus 10 m/s, then minus 20 m/s, then minus 30 m/s, and so on. Even though the potato's *speed* was increasing, the *velocity* was becoming more negative by 10 m/s each second. The potato's change in velocity with time was negative 10 m/s each second, or negative 10 m/s^2.

Speed 4 TAP

So what negative acceleration really means isn't "speeding up" or "slowing down".

It means that the change in velocity with time is negative. Whatever your velocity is now, you get the velocity one second later by adding a negative number to what you already had.

Speed 4 TAP

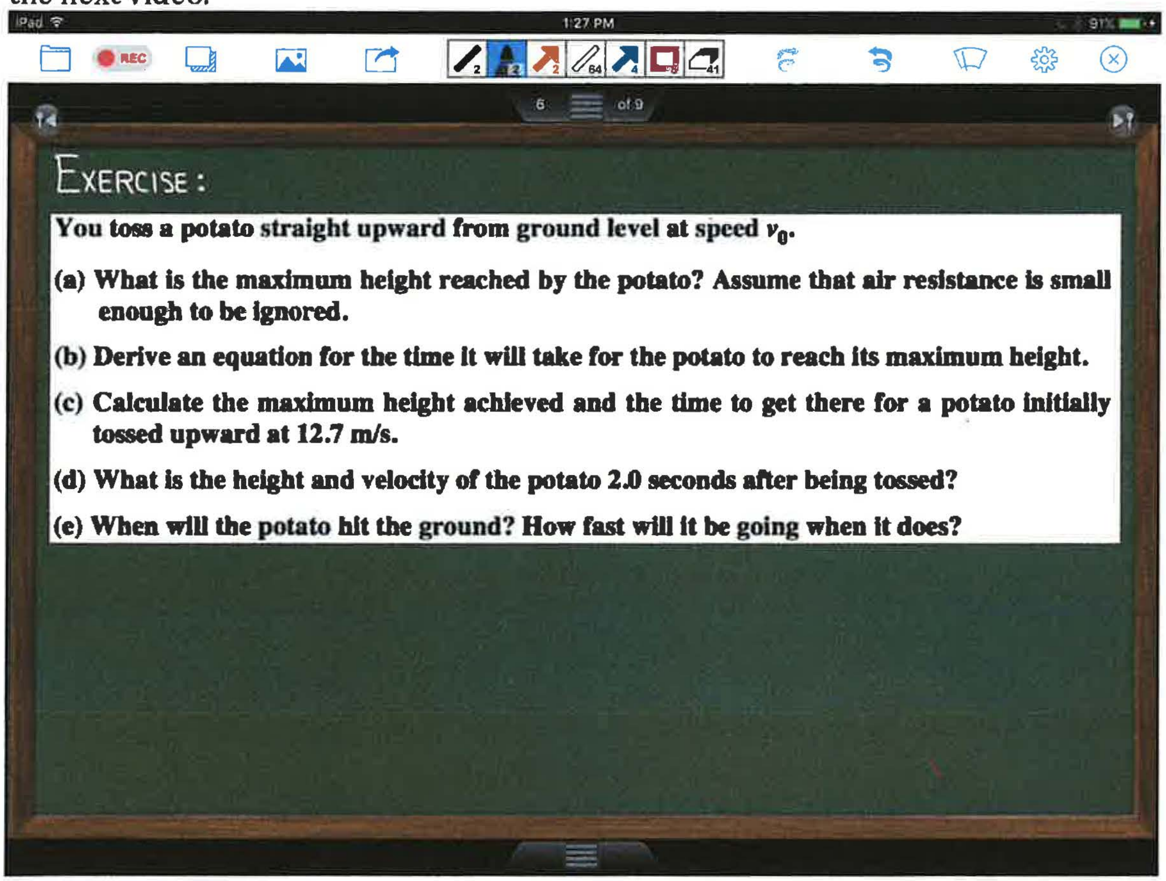
If you think of the velocity as changing along a number line, each second the velocity moves 10 m/s along the line in the negative direction. ON the way up, that means the velocity it become less positive—the potato is slowing down.

TAP

On the way down the velocity is becoming more negative—the potato is speeding up.

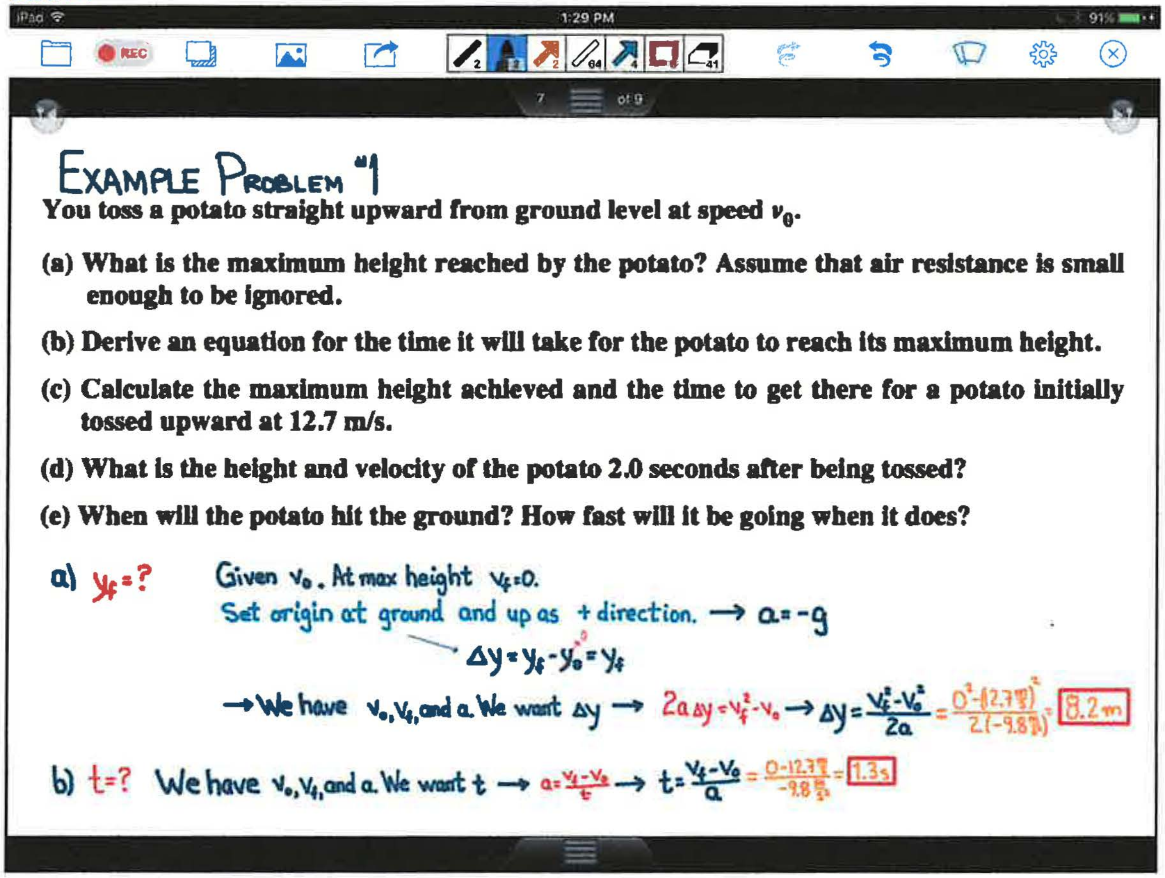
Slide 6

Here's an exercise to try on your own. Give it a shot, and we'll look at the solution in the next video.



Begin new segment

Slide 7



Tap to next stop

Greetings! Welcome back.

We we're looking at an example problem of a potato tossed straight up from the ground with an initial upward speed v zero, and we wanted to find various things about its time in the air.

Part a asks us to find its maximum height, which I'll call y final.

(TAP)

We're given v zero. We know that at the maximum height v final equals zero.

(TAP)

We always need to choose an origin and positive axis direction. I'll put the origin on the ground and call upward the positive direction, which means that our acceleration, which is downward, will be negative g.

(TAP)

In this case our displacement delta y is yfinal minus y initial, which is just y final because our initial height is zero.

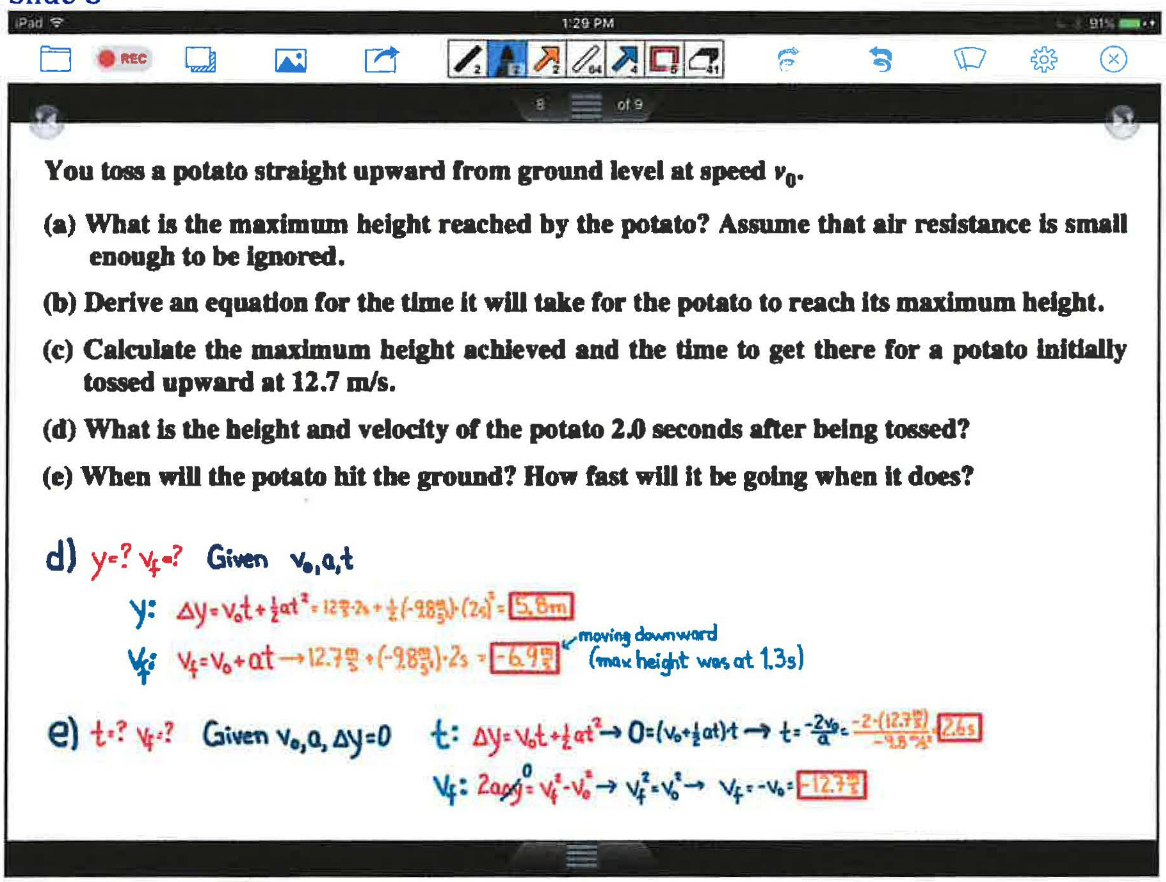
We have v zero, v final, and a, and we want delta y.

(TAP)

So we'll choose the equation 2 a delta y equals v final squared minus v initial squared and rearrange to get our expression for delta y.

Part b asks us to find the time to reach the maximum height.
(TAP)
Here we know v zero, vfinal and a and want t
(TAP)
We can use our defining equation for acceleration and solve it for t.
Part c gives us the initial speed of 12.7 meters per second, and then asks us to plug in numbers
(TAP)
Done with that part.

(TAP)
Slide 8



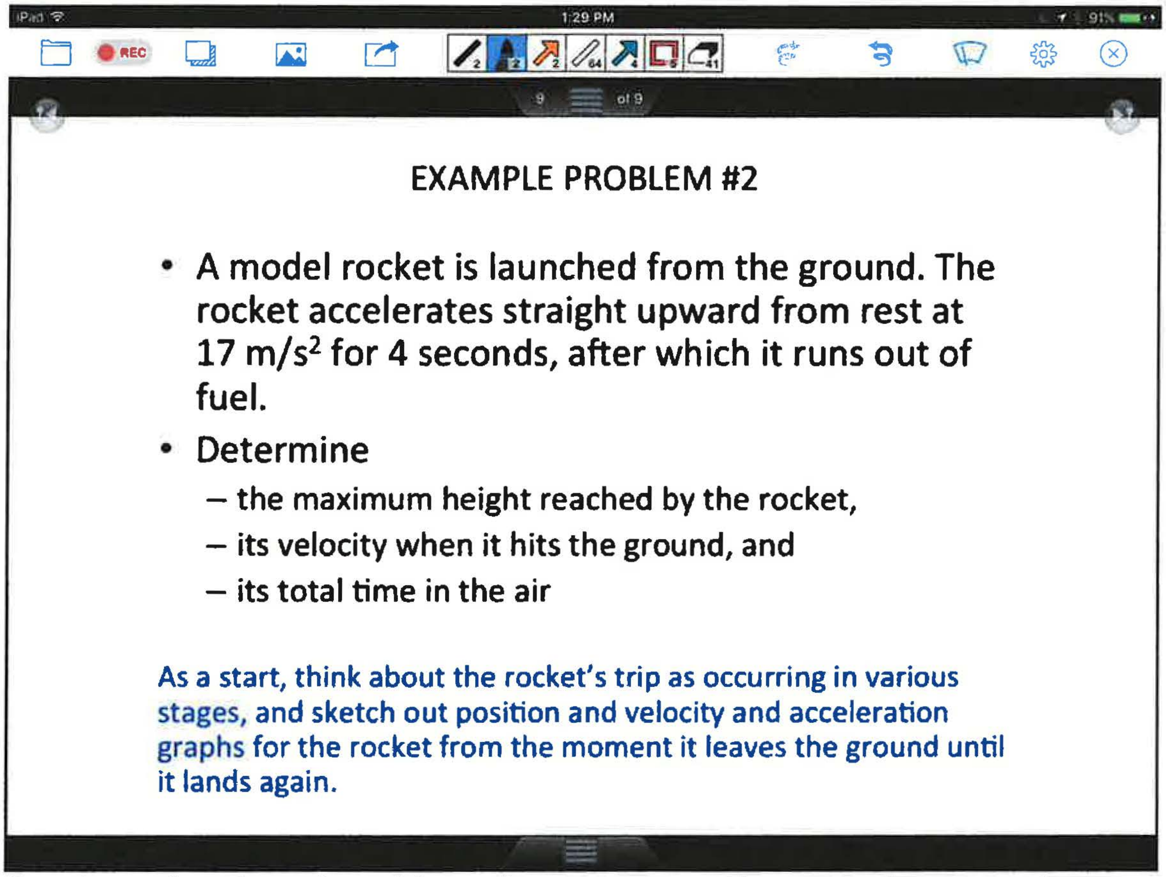
Part d asks us to find the height and velocity two seconds after the toss. We're given v zero, a, and t and asked to find delta y and v final. Here goes.
(TAP)
To find y we use the equation delta y equals v zero t plus one half a t squared.
To find v final we can us v final equals v zero plus a t.
How do I know which equation to use? For constant acceleration problems there are only five equations, which we have seen earlier. Each involves four variables, and your are always given three quantities and are asked to find the fourth one. So it's really "equation shopping" rather than physics. You go through them until you find the one that works for you. If you've memorized them, this is a pretty quick process.

Back to the solution . . . we plugged in our numbers and got that the potato was 5.8 m above the ground and going 6.9 meters per second downward, which makes sense, because we figured out in part c that the potato already reached its maximum height at 1.3 seconds.

The final part of the problem asks us to find the time when the potato hits the ground, and how fast it is going when it gets there.
(TAP)
We're given v zero, a, and we know that delta y equals zero. We can choose appropriate equations and then solve.
The two useful results from this are that, in the absence of air resistance, the trajectory of a thrown object is symmetric. It takes just as long to come down as it does to go up, and the speed when it hits the ground is the same as when it was tossed.

(TAP)

Slide 9

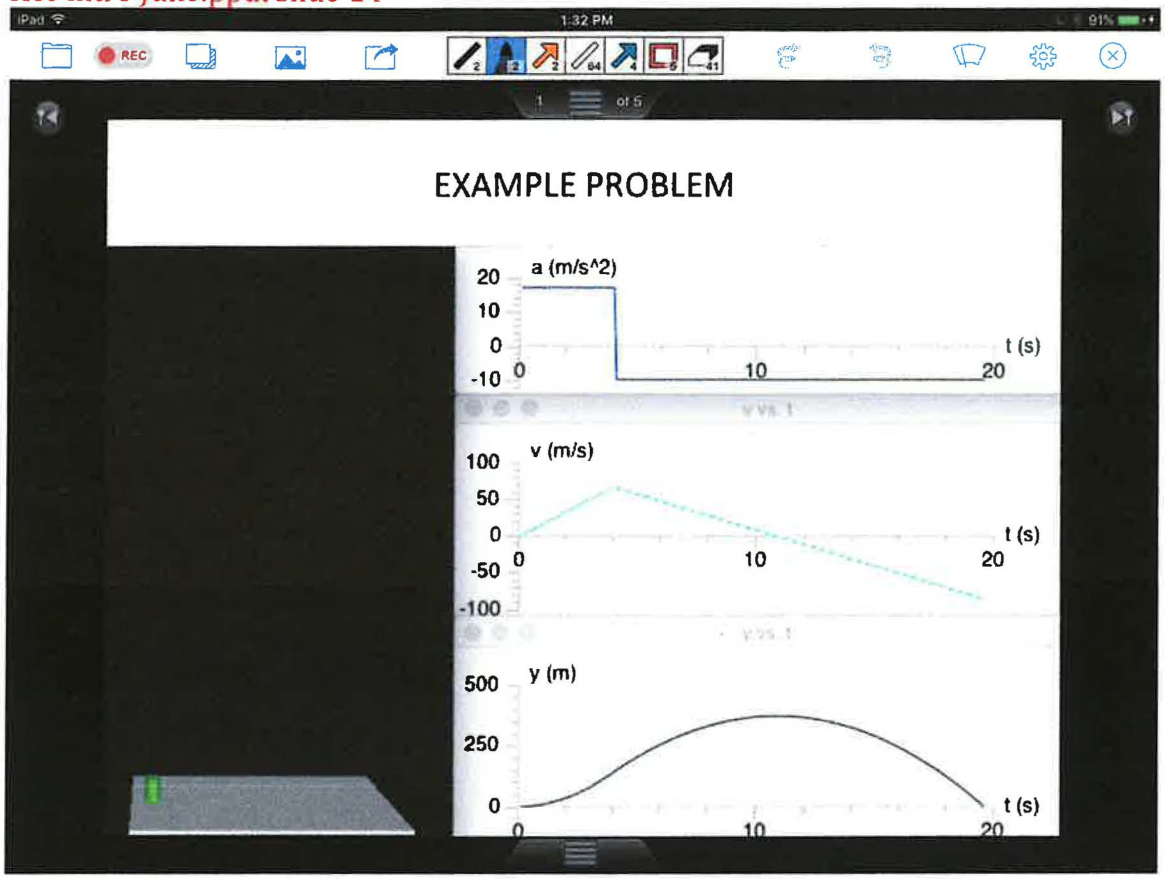


Here is another Example problem, where we can put these equations to work:

- A model rocket is launched from the ground. The rocket accelerates straight upward from rest at 17 m/s^2 for 4 seconds, after which it runs out of fuel.
- Until it hits the ground, its acceleration is negative 9.8 m/s^2. Assume that air resistance is not a factor.
- We want to determine
  - the maximum height reached by the rocket,
  - its velocity when it hits the ground, and
  - its total time in the air

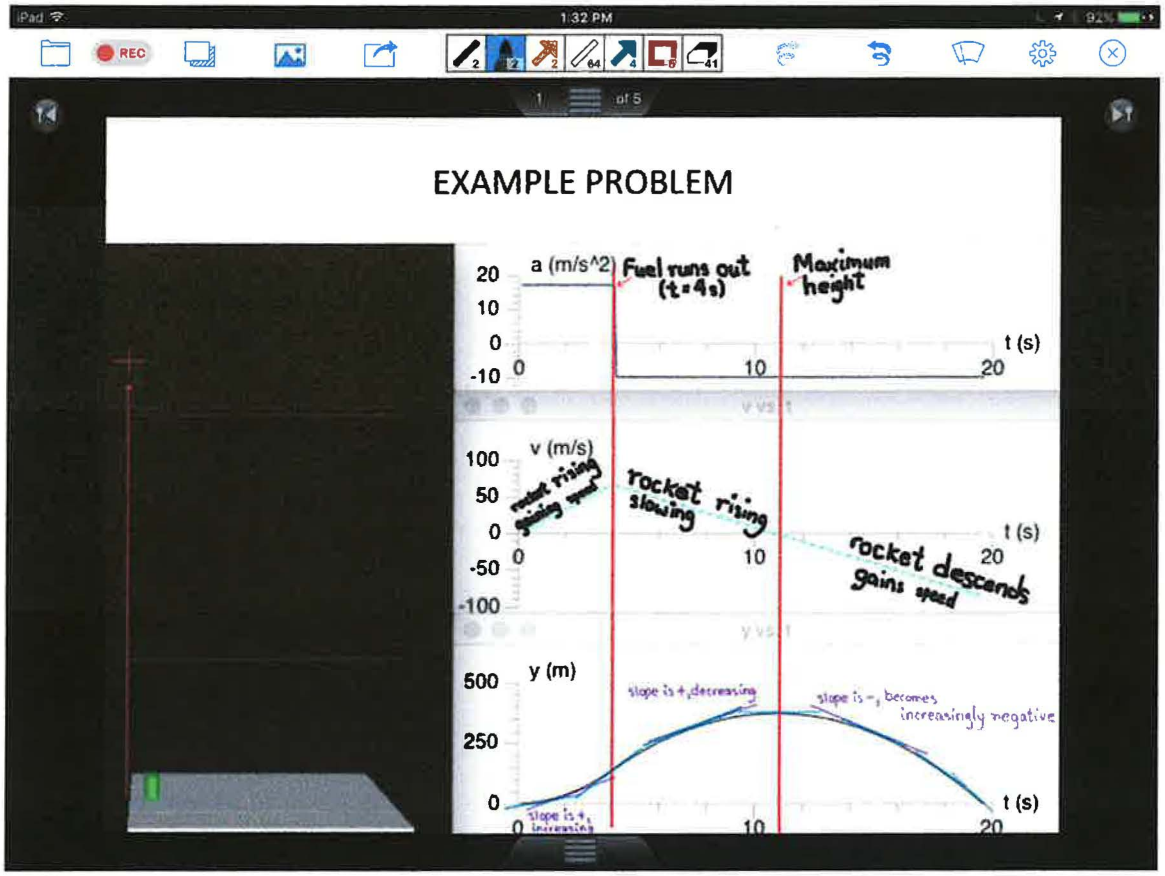
For a start, it would be helpful to think about the rocket's trip as happening in various stages or parts, and to sketch out the acceleration, velocity, and position graphs. Sketch out these three graphs for the motion of the rocket, from the instant it leaves the ground until the moment it lands again. I'll see you in the next video.

Acc intro June.pptx slide 14



Greetings! Welcome back.
We were looking at a rocket problem and wanted to generate the acceleration, velocity and time (position) graphs for the rocket's path. Here's what they look like. You can see that the rocket accelerates for the first four seconds, runs out of fuel, and then gravity takes over.

Cool. Let's look at the graphs individually.



TAP

On the velocity graph, you can see that the velocity increases steadily for the first four seconds. This is matched on the acceleration graph, which shows a constant positive (or upward) acceleration

TAP

At t = 4 seconds the rocket runs out of fuel. Now gravity is the only force acting and the acceleration is 9.8 meters per second squared downward. The velocity is still positive (the rocket is still rising) but decreased to zero when the rocket reaches its maximum height around t equals 11 seconds.

TAP

From there the velocity becomes increasingly negative as the rocket accelerates toward the ground.

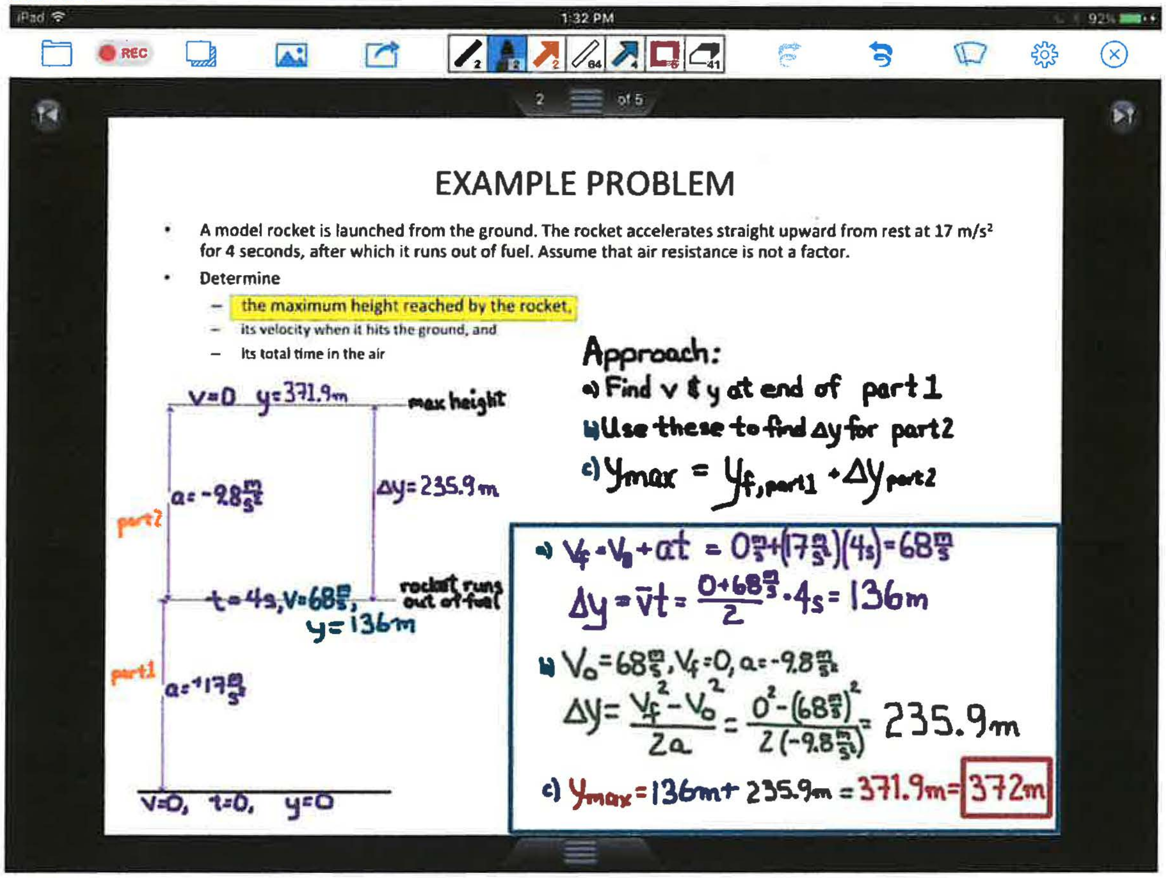
We can get the same sense of what is going on by looking at slope of the position graph, in this case the height as a function of time.

The slope starts out at zero, is positive and increasing until t = 4 seconds, when the rocket runs out of fuel; it's positive and decreases to zero when the rocket reaches its maximum height; and then the slope becomes increasingly negative as the rocket descends and gains speed.

(TAP)

(rocket prob sol part 1)

Slide 2



Let's get back to working out this problem:

The first part asks us to find the maximum height reached by the rocket.

(TAP)

Our analysis from the graph tells us that there are two separate parts to the rockets rise. For the first four seconds its acceleration is 17 meters per second squared, and for the rest of the rise the acceleration is minus 9.8 meters per second squared.

(TAP)

So our approach will be to find the speed and height at the end of four seconds, then use that to find the maximum height, where we know that v final will be zero.

Here goes:

(TAP)

For part 1 we know v zero, a, and t, and can use our constant acceleration kinematics equations to find that the velocity after 4 seconds is 68 m/s and the height is 136 meters.

(TAP)

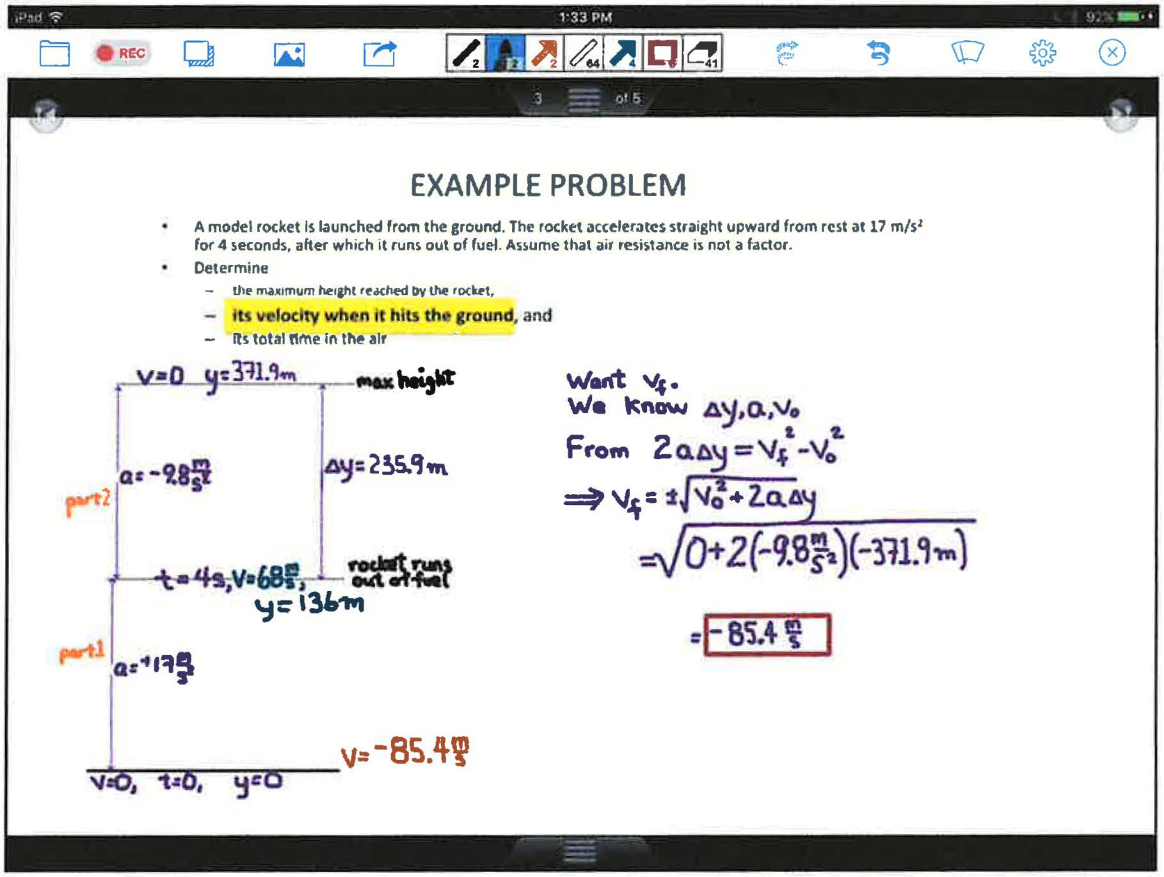
For part 2 we know that v initial is 68 m/s, v final is zero, and a is -9.8 meters per seconds squared.

We solve to find that the rockets additional rise is 235.9 meters, for a total rise of 371.9, or 372 meters. Hooray!

(TAP to next stop)

(rocket prob sol part b)

Slide 3



For part b we want to find the velocity when the rocket hits the ground.
We know delta y (which is negative 371.9 m—*negative* because we called upward the positive direction, and here y final is less than y initial), we know v zero (which is zero) and we know the acceleration (minus 9.8 meters per second squared).

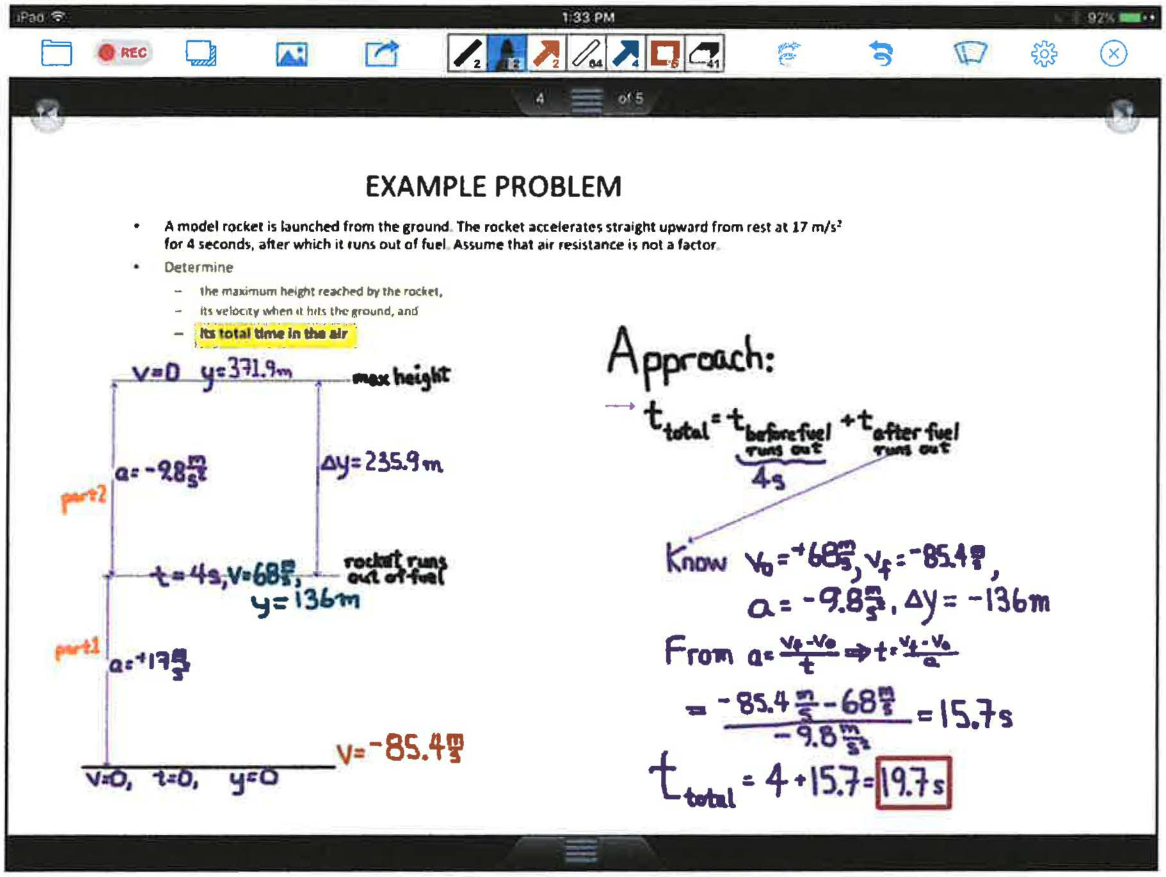
(TAP)

We choose an appropriate equation—one that has delta y, a, v initial, and v final in it—plug in, and get our result of negative 85.4 meters per second.

(TAP to next stop)

(rocket prob sol part c)

Slide 4



For part c we want to find the total time the rocket is in the air. We'll break this up into two parts—the time while the rocket is accelerating and burning fuel, and then the time after the rocket runs out for fuel.

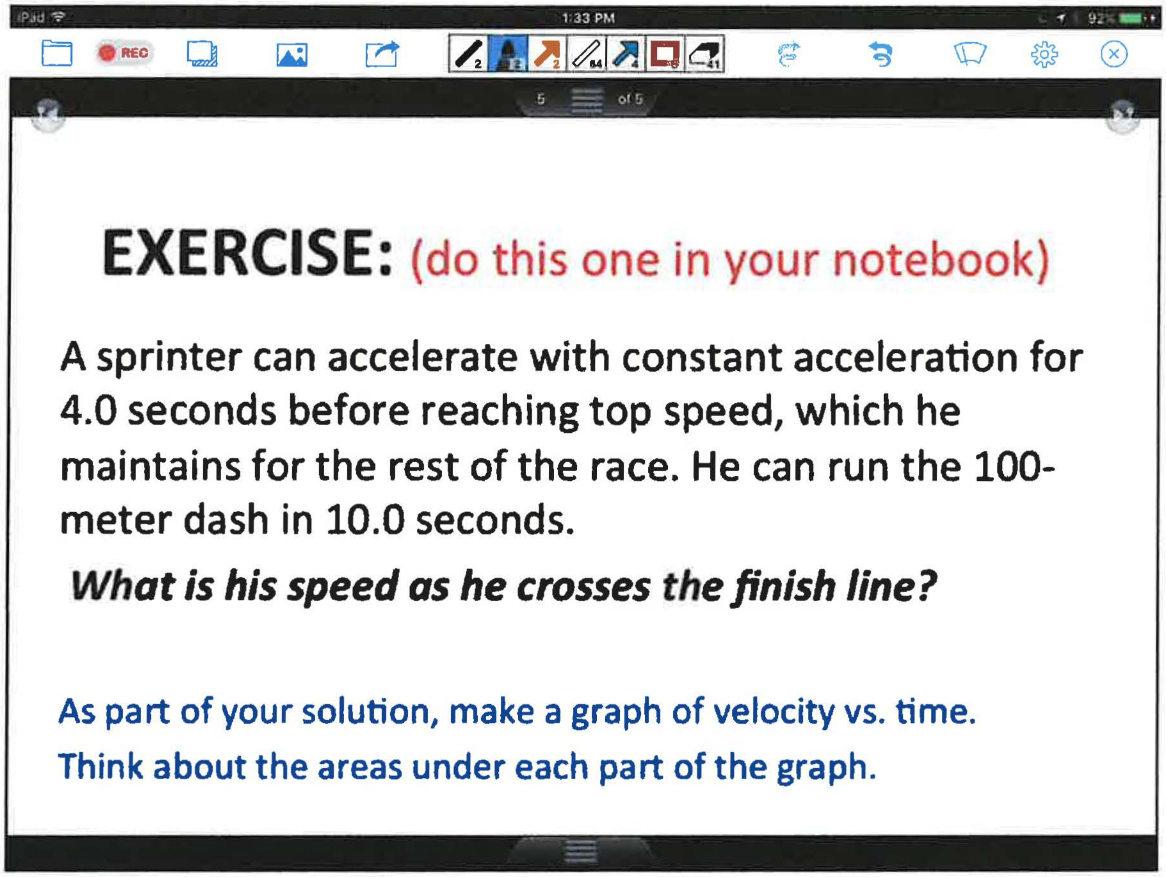
The time for the first part is four seconds—this was given in the problem. For the second part we know the initial velocity v zero is plus 68 meters per second, the final velocity (which we just found to be negative 85.4 meters per second, the acceleration (-9.8 m/s^2), and delta y (which is negative 136 m).

We could choose just about any equation with a t in it.

TAP

I chose one. It gives 15.7 seconds for the second part of the flight for a total of 19.7 seconds that the rocket is in the air.

TAP:



Here's an exercise to do on your own in your notebook:

A sprinter can accelerate with constant acceleration for 4.0 seconds before reaching top speed, which he maintains for the rest of the race. He can run the 100-meter dash in 10.0 seconds.

What is his speed as he crosses the finish line?

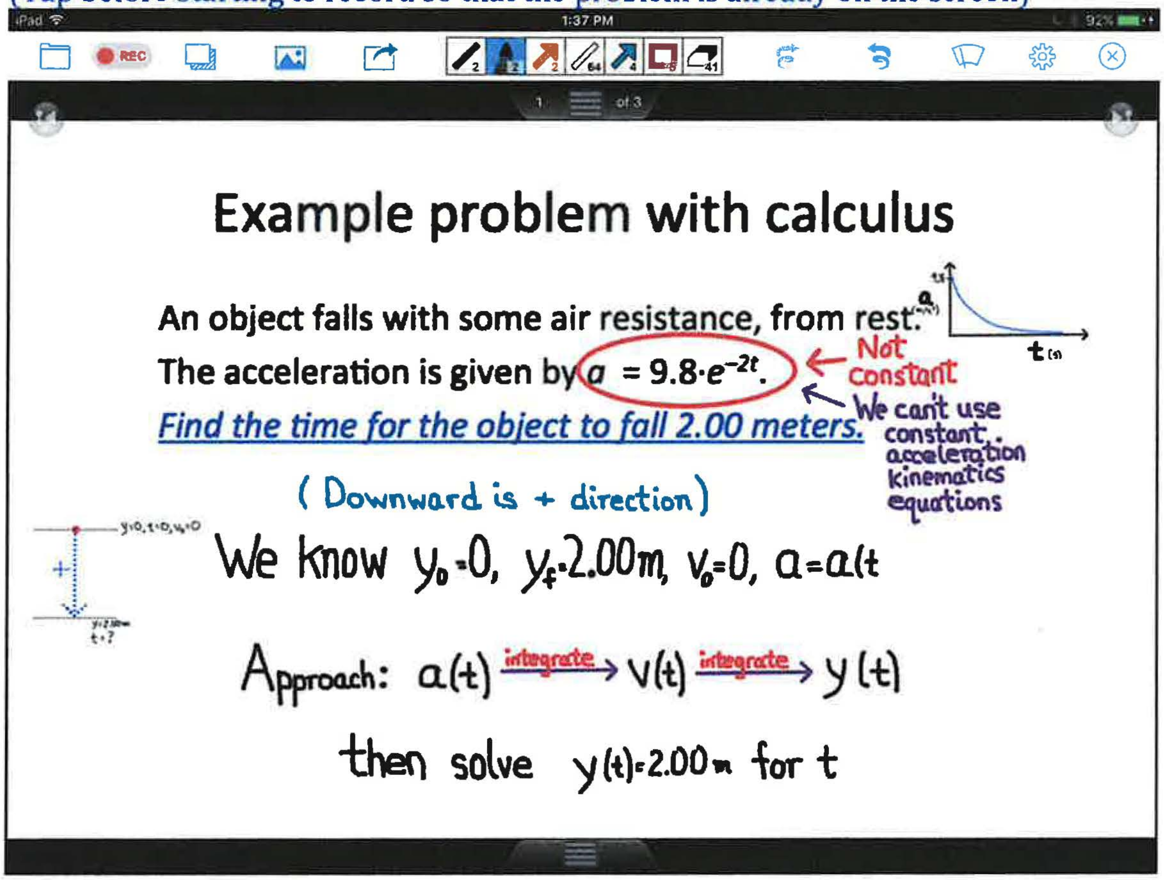
Make a graph of velocity vs. time representing his entire run.

As in the previous problem, you can consider the problem as occurring in two different stages.

And, as in the previous problem, the velocity at the end of the first stage is the velocity at the beginning of the second stage. (You don't know this velocity initially, but you can write an expression for it.)

Give this one a shot in your notebook and I'll see you in the next segment.

Begin new segment
Doceri example acc prob calc1
Slide 1
(Tap before starting to record so that the problem is already on the screen)



Here is an example problem involving calculus.
An object falls with some air resistance, starting from rest.
The acceleration of the object is given by a of t equals 9.8 times e to the negative 2t.
Find the time for the object to fall 2.00 meters.

So, we're looking for the time.
TAP
Notice that the acceleration is NOT constant! At t equals zero the acceleration is 9.8 m/s^2 downward, but this acceleration decreases as time goes on due to air resistance, which builds up the faster the object falls.
This means that we can't use our constant acceleration kinematics equations.
We'll have to use calculus.
TAP
There are some things we do know.
We can call our initial position zero, and call our final position 2.00 meters (so we're calling downward the positive direction). Since the object starts from rest, its initial velocity is zero, and the acceleration is a function of time.

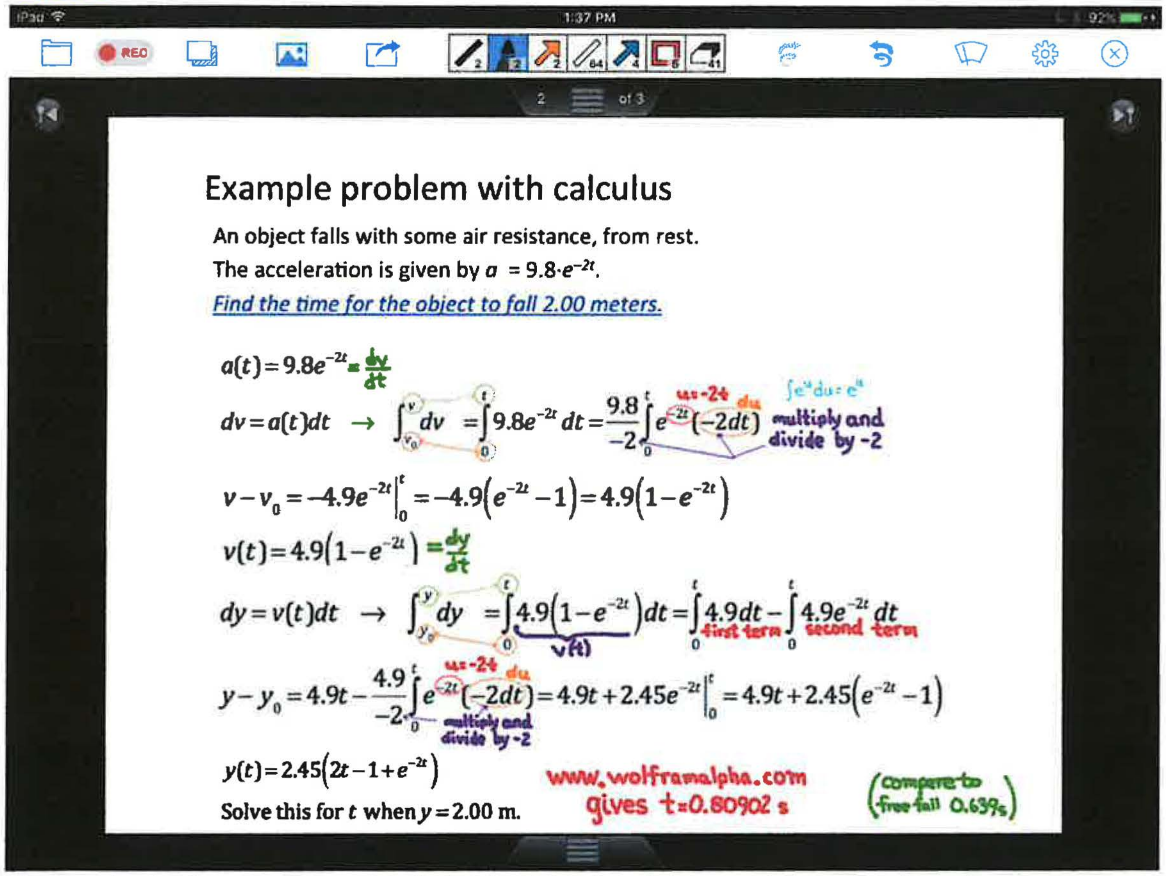
TAP

A way to approach this problem is to integrate the expression for acceleration to get velocity as a function of time, integrate *that* expression to get position as a function of time, and then set this expression equal to 2.00 meters and solve it for t.
Here we go!

We start with our expression for the acceleration, and then integrate it.

(TAP)

Slide 2



Since a is dv dt, we can multiply both sides by dt to get dv = a dt, then integrate. These are definite integrals. We integrate time from zero to t, and we integrate velocity from v zero (the velocity that corresponds to t = 0) to v (the velocity at time t).

So the integral of dv from v zero to v equals the integral of our acceleration, 9.8 times e to the negative 2t, from zero to t.

(TAP)

I want to do a u-substitution, so I set -2t equal to u, and then du is -2dt, so I multiplied and divided by -2 to make the du term look right.

(TAP)

When we do the integral we get v – v0 = -4.9 times e to the negative 2t, evaluated at zero and t. Doing that, and setting v0 = 0 gives our expression for the velocity as a function of time. V(t) = 4.9 times the quantity (1 minus e to the -2t)

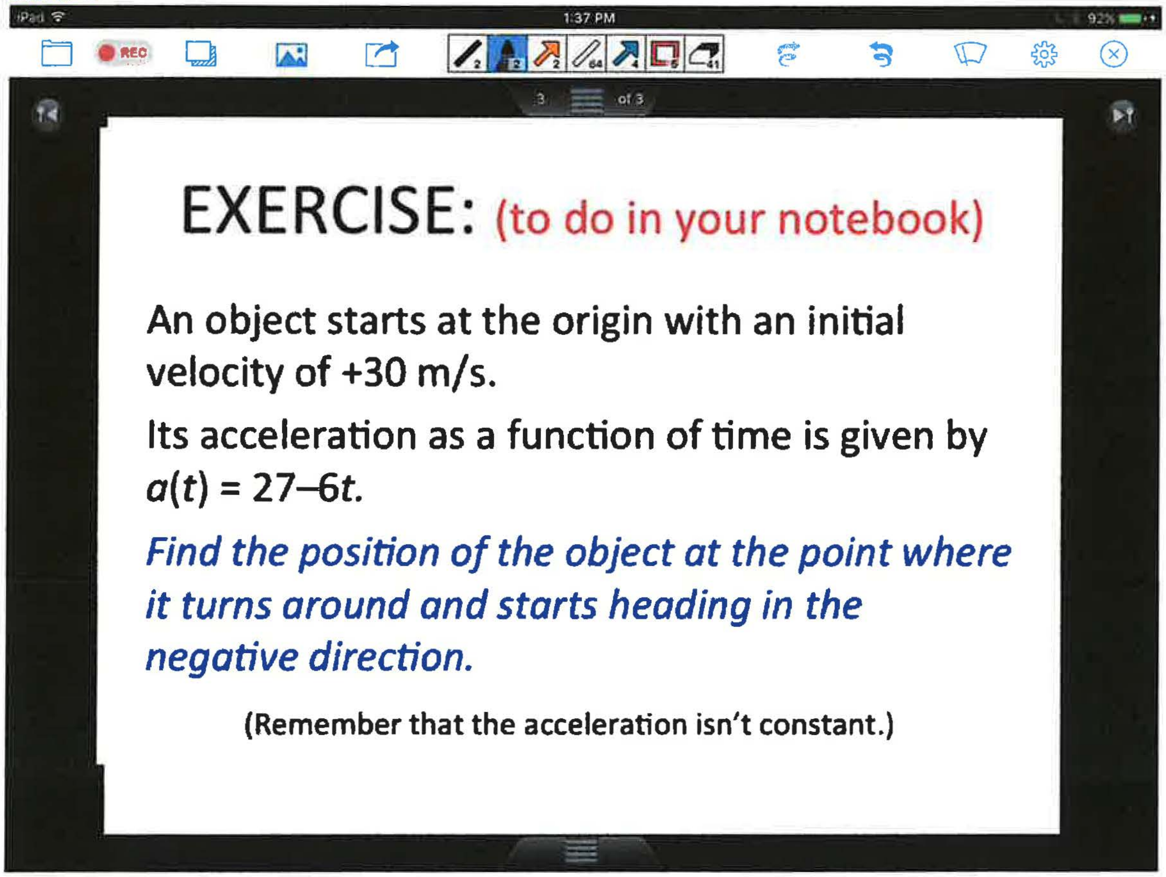
An interesting thing to notice is that as t goes to infinity the velocity becomes 4.9 m/s. That will be the object's terminal velocity

Now we need to integrate the velocity function to get the position as a function of time. Here goes.
(TAP)
V of t is dy dt. We can multiply both sides by dt and then integrate.
Again, these are definite integrals. Time runs from zero to t, and position runs from y-zero to y.
The integral of dy from y-zero to y equals the integral from zero to t of 4.9 times (one minus e to the negative 2t) dt. I split this last expression into two terms so that I could integrate each one separately. The first term gives 4.9 t, and we use the same u-substitution from before on the second term. I've worked out the integral.
(TAP)
It gives y of t equals 4.9 t plus 2.45 times e to the -2t minus 1.

(TAP)
This can be rearranged a bit, to y of t equals 2.45 times the quantity (2t minus 1 plus e to the -2t). We can set this equal to 2.00 and then solve for t.

This turns out to be a difficult thing to solve. Perhaps you have a solve function on your calculator.
(TAP)
My favorite approach is to go to wolframalpha.com and ask them to solve it. That gives t = 0.80902 seconds as the time to fall 2 meters. The free fall time to fall two meters is .639 seconds, so it's at least reassuring that air resistance makes it take longer to fall the same distance.

Slide 3



TAP

Here's an exercise to do in your notebook.

Since the acceleration isn't constant, you need to use calculus. I've chosen the numbers so that the algebra shouldn't be too bad.

Do well!

Rocket Science

Rocket Science lecture videos—Script and Screenshots

The following pages contain the script for the video lectures introducing a calculus-based approach to deriving equations for rocket propulsion. They were recorded into movies.

The "slides" themselves printed herein represent screenshots from Powerpoint (in which case they involve animations that won't show up in a static screenshot) or Doceri (in which case the text and images on the actual screen appear as a sequence of pen strokes or images on the screen, appearing one at a time). Each screenshot included here is the *final* version of what appears on the screen right at the end of that particular slide. Although these give a sense of what the lecture looks like, best would be to watch the actual lectures themselves. (Links to those are included elsewhere in this report.)

Because I scripted the voiceovers, various words appear in the script to tell me when to advance to the next animation or set of strokes on the screen ("Tap" or Click"); how fast to set the rate at which things appear on the Doceri screen ("Speed 8"); or when to advance through a whole set of strokes all at once in Doceri instead of things appearing one stroke at a time ("Tab to next stop").

In addition, each lecture is a compilation of various slides, some from Powerpoint and some from Doceri, so there are notes at the beginning of some slides as to what presentation or Doceri project to start the narration from.

As a whole, this set of lecture videos comprises five videos, about 24 minutes total:

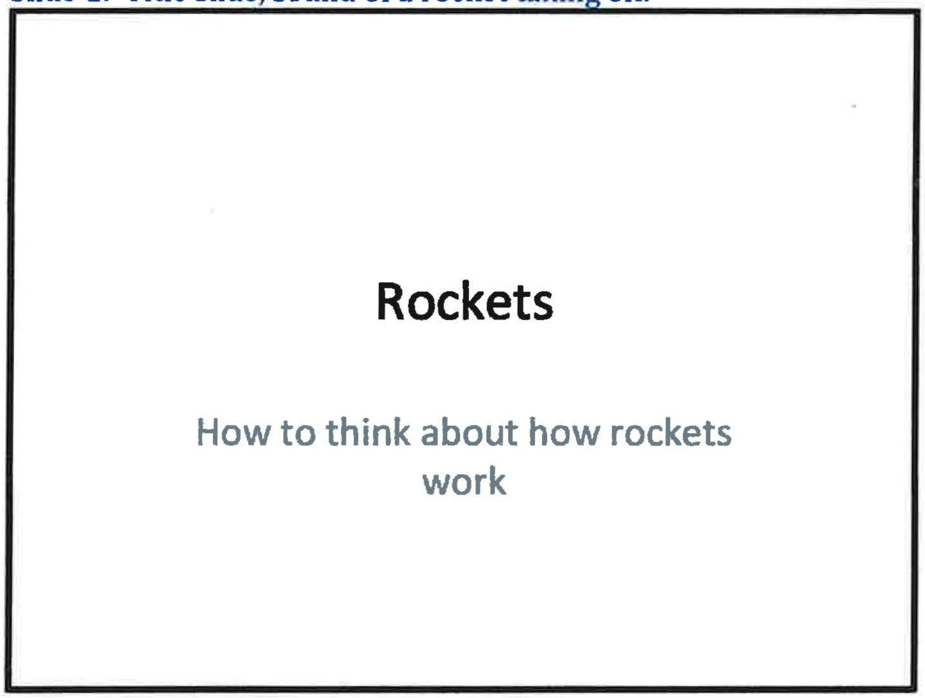
Title	Length (minutes:seconds)
Rocket momentum part_1	3:00
Rocket momentum part_2	7:58
Rocket momentum part_3	6:31
Rocket momentum part_4	3:10
Rocket momentum part_5	3:22

Links for students to access the videos are posted in Moodlerooms. I also post a student version of the script, which is essentially all of the text in what follows but without any of the "Tab" or "Click" instruction. The student version is separated out by lecture movie title rather than by the name of the source file.

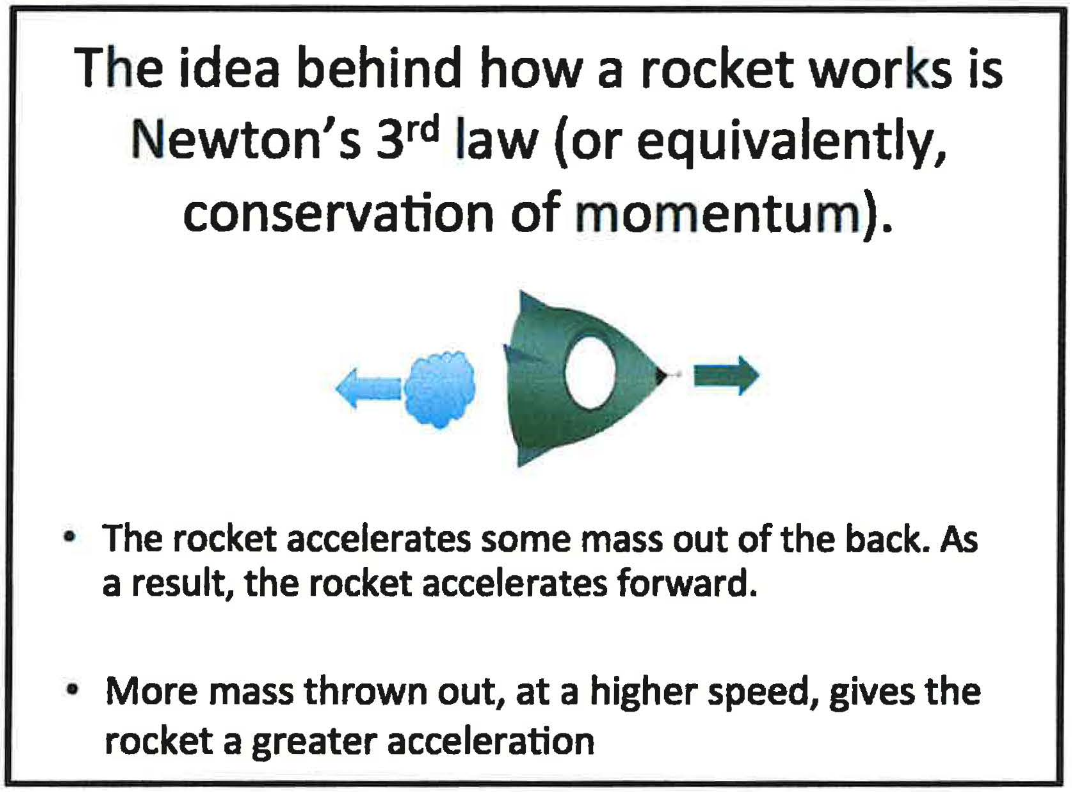
The approach here is to start with the discrete case of finding the change in the velocity of a railroad flatcar, initially at rest, when you throw a mass off of the back of the car at some speed relative to the car. Then to generalize this to breaking that mass up into two, or three, or a million pieces and then throwing each piece off, one at a time. Then finally to extend this to the continous case, where we imagine breaking that mass up into an infinite number of infinitely small pieces and then throwing them off the flatcar, one at a time (calculus!)

Rocket intro2.pptx

Slide 1: Title slide, sound of a rocket taking off.



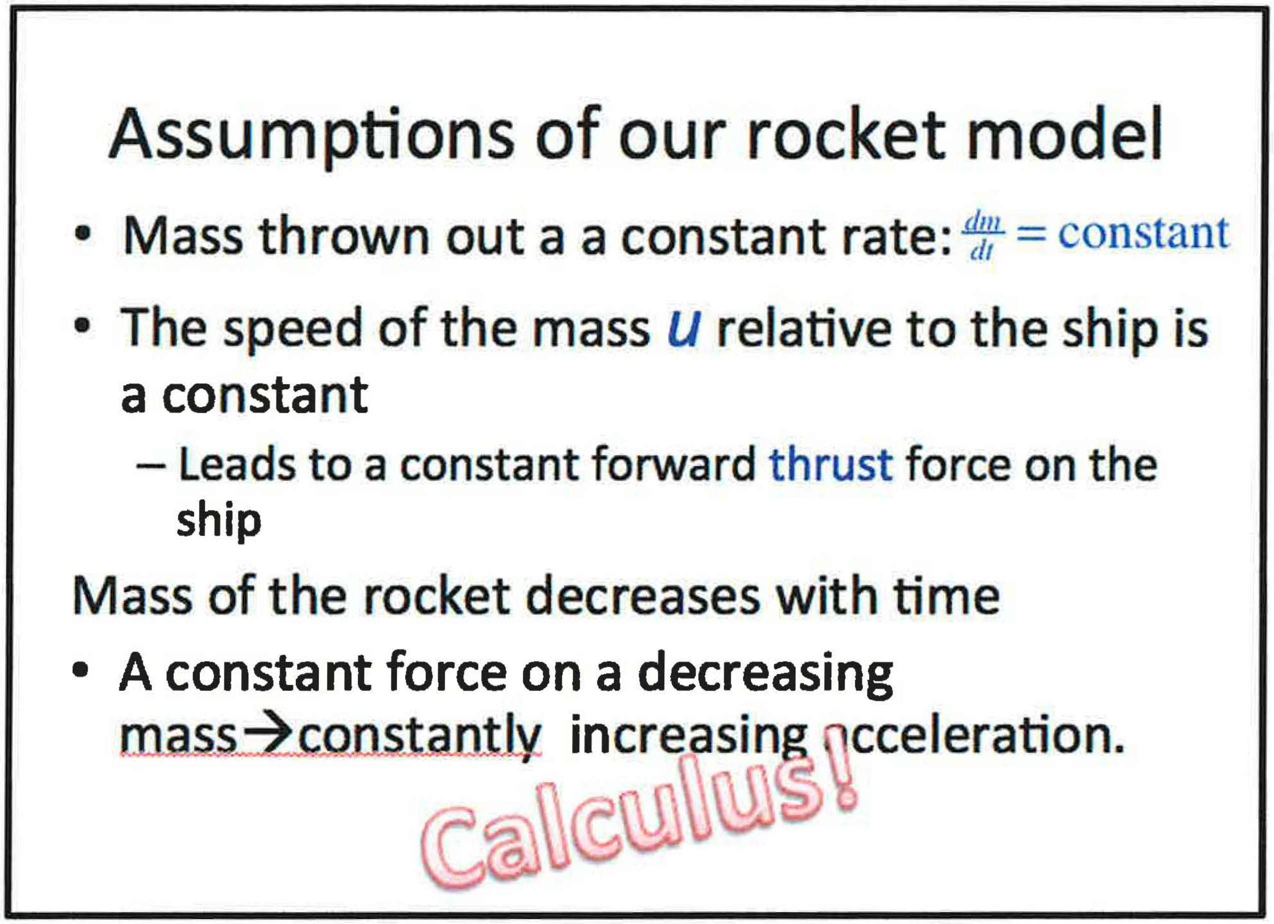
Slide 2



The idea behind how a rocket works is Newton's 3rd law (or equivalently, conservation of momentum).
Click The rocket accelerates some mass out of the back. As a result, the rocket accelerates forward.
Click (animation)

If you keep throwing mass out the back, the rocket *keeps* accelerating forward.
Click
The more mass you throw out the back, and the faster you throw it, the faster the rocket goes.

Slide 3



Click

For our simplest model of a rocket, we'll assume that the mass is thrown at a a constant rate ($\frac{dm}{dt}$=constant) and

Click

that the speed u of the exhaust or propellant is also constant *relative to the rocket.* This will give us a constant force backward on the exhaust, and a constant *thrust* force forward on the rocket.

Click

Since the rocket is constantly throwing mass out the back, the mass of the rocket is decreasing with time, and so the acceleration of the rocket due to this constant thrust force will be *increasing* with time, in a smooth continuous way.

Click Calculus! Hooray!

Click

Slide 4

Our approach . . . .

- Start with the discrete case
  - Imagine throwing bits of mass off the rocket, one at a time
- Move to the continuous case
  - Chunks of mass being thrown out the back get very small, and come one right after the other

We'll approach the rocket problem by first examining throwing fuel out of the rocket in discrete chunks, one bit at a time, and then extend this to throwing much smaller bits of mass out the back, one right after the other, in a continuous stream.
SO: First the discrete case.

Slide 5

Imagine a flatcar, mass M, at rest.
On it, two objects, each of mass m.
Thrown at speed u relative to flatcar

EXERCISE: If you throw both masses off at the same time at speed u relative to the flatcar, show that the final speed of the flatcar relative to Earth is

$$v_f = \frac{2m}{M+2m}u$$

$P_{system,0} = 0$ relative to Earth. Write all final velocities relative to Earth. (u is not given relative to Earth.)

Imagine that you are standing on a flatcar at rest.
The mass of you and the flatcar together is *M*.

On the flatcar are two objects, each of mass *m*.
You can throw any mass off of the back of the flatcar at a speed *u* relative to the car.
Click
If you throw them both off at the same time, conservation of momentum gives that the final speed of the flatcar is two *m* divided by the quantity (capital *M* of the flatcar plus two *m*), all times *u*, the speed of the thrown mass relative to the flatcar :

$$v_f = \frac{2m}{M+2m}u$$

The EXERCISE is to:

Derive this equation for v_f of the flatcar when you throw both masses off at the same time at speed *u* relative to the flatcar.

Click
The trick here is that the initial momentum of the system is zero in the frame of reference of the Earth. So when you set up your conservation of momentum equations, you'll want to write the expressions for the final momentum of the flatcar and of the two masses using their velocities relative to Earth.
Try this one on your own. I'll see you again in a couple of minutes.

Slide 6

EXERCISE:

- If you throw both masses off at the same time at speed *u* relative to the flatcar, show that the final speed of the flatcar is

$$v_f = \frac{2m}{M+2m}u$$

SOLUTION: Apply conservation of momentum to the flatcar + masses system
Solve in the reference frame of the Earth.

$$v_{A/C} = v_{A/B} + v_{B/C}$$

$$p_{before} = p_{after}$$

$$0 = Mv_{flatcar/Earth} + 2mv_{2m/Earth} = Mv_{flatcar/Earth} + 2m(v_{2m/flatcar} + v_{flatcar/Earth})$$

$$0 = (M+2m)v_{flatcar/Earth} + 2mv_{2m/flatcar} = (M+2m)v_{flatcar/Earth} + 2m(-u)$$

$$\Rightarrow \boxed{v_f = v_{flatcar/Earth} = \frac{2m}{M+2m}u}$$

Greetings! Welcome back.
We were looking at a problem where you are standing on a flatcar, with total mass capital *M*. Also on the flatcar are two smaller masses lower case *m*. We wanted to calculate the change in the speed of the flatcar if you throw both small masses *m* off the car at the same time at a speed *u* relative to the flatcar.

Click
For our SOLUTION, we apply the law of conservation of momentum to the system of the flatcar and the two masses: We'll solve this problem in the reference frame of the Earth, where the initial momentum of the system is zero, and is equal to the final momentum.

Click
So we have p before equals p after.
Zero equals the mass of the flatcar times the velocity of the flatcar plus the thrown mass 2*m* times the velocity of the thrown mass, all relative to Earth.

Next we rewrite our equation, but for the velocity of the 2*m* relative to Earth we write the sum velocity of 2m relative to the car, plus the velocity of the car relative to Earth.

Click

This is just like v of A relative to C is v of A relative to B plus v of B relative to C.

Click

Then we gather like terms, substitute in that the velocity of the 2*m* relative to the flatcar is $-u$, and solve for the final velocity of the cart, and get the desired result.

$$0 = Mv_{\text{car/Earth}} + 2mv_{\text{2m/Earth}} = Mv_{\text{car/Earth}} + 2m(v_{\text{2m/cart}} + v_{\text{cart/Earth}})$$

$$= (M+2m)v_{\text{cart/Earth}} + 2mv_{\text{2m/cart}} = (M+2m)v_{\text{cart/Earth}} + 2m(-u)$$

$$\Rightarrow v_{\text{f}} = v_{\text{cart/Earth}} = \frac{2m}{M+2m}u$$

Slide 7

Suppose instead that you throw first one mass off, and then the other.

$v_{m/Earth} = v_{m/car} + v_{car/Earth}$

$= -u + v_{f1}$

$$0 = (M+m)v_{f1} + m(-u + v_{f1}) = (M+2m)v_{f1} - mu$$

After the first mass you have:

$$\Rightarrow v_{f1} = \frac{m}{M+2m}u$$

We'll now toss the second mass off the back, but *in the frame of reference of the flatcar*, so that the total momentum in that frame is zero:

$$0 = Mv_{f2} + m(-u + v_{f2}) = (M+m)v_{f2} - mu$$

$$\Rightarrow v_{f2} = \frac{m}{M+m}u$$

The total change in speed of the car (in the frame of the Earth) after throwing the two masses off, one at a time, is :

$$\Delta v_{car} = v_{f1} + v_{f2} = \frac{m}{M+2m}u + \frac{m}{M+m}u$$

Suppose instead that you throw first one mass off, and then the other.
In the reference frame of the Earth, after you throw off the first mass, the mass left behind is that of the flatcar plus little m,

Click

moving at a velocity I'll call v final 1, or v f 1.

Click

The velocity of the thrown mass relative to Earth is the velocity of the thrown mass relative to the car plus the velocity of the car relative to Earth, which is minus u + vf1

When we write our conservation of momentum equations we get:

Zero equals the mass of the flatcar plus the little *m* still on the car, times the final speed of the car, plus the thrown mass *m* times its velocity relative to Earth, which is its velocity relative to the cart (-u) plus the velocity of the cart relative to Earth (v_{f1}).
We can collect terms and then solve for v_{f1}, the speed after the first mass is thrown.

Click

We'll now toss the second mass off the back, but in the frame of reference of the flatcar, so that the total momentum in that frame is zero:

Again, the initial momentum is zero, and the final momentum is the mass capital M of the car times its velocity, plus the thrown mass little *m* times its velocity relative to this frame of reference, which is its velocity relative to the car (-u) plus the velocity of the car relative to this frame (v_{f2}).

We can collect terms and then solve for v_{f2}, the speed in this frame of reference after the first mass is thrown.

$$0 = Mv_{f2} + m(v_{f2} - u) = (M + m)v_{f2} - mu$$

$$\Rightarrow v_{f2} = \frac{m}{M+m}u$$

Click

The total change in speed (in the frame of the Earth) after throwing the two masses off, one at a time, is the sum of these two changes in velocity, v_{f1} and v_{f2}.

$$\Delta v = v_{f1} + v_{f2} = \frac{m}{M+2m}u + \frac{m}{M+m}u$$

Slide 8

Compare these two cases where *M* is 10 times each *m*.

Tossing both masses at the same time gives

$$v_f = \frac{2m}{10m+2m}u = \tfrac{1}{6}u = 0.167u$$

while tossing them one at a time gives

$$v_f = \frac{m}{M+2m}u + \frac{m}{M+m}u = \frac{m}{10m+2m}u + \frac{m}{10m+m}u = \left(\tfrac{1}{12}+\tfrac{1}{11}\right)u = 0.1742u$$

(This second Δ*v* is faster because the reaction mass when you toss the second mass separately is a little bit less, so the acceleration of the flatcar during the throw is a little bit more.)

Let's compare our two cases for the mass of the flatcar *M* being 10 times each little mass *m*. Tossing both masses at the same time gives
V final equals two m divided by 10 m + 2m times u, which gives one-sixth u, or 0.167 u

$$(v_f = \frac{2m}{10m+2m}u = \tfrac{1}{6}u = 0.167u)$$

while tossing them one at a time gives
m over ten m + two m times u , plus m over ten m + one m times u, which gives one-twelfth plus one-eleventh, which is 0.1742u.

$$(\Delta v = \frac{m}{M+2m}u + \frac{m}{M+m}u = \frac{m}{10m+2m}u + \frac{m}{10m+m}u = \left(\tfrac{1}{12}+\tfrac{1}{11}\right)u = 0.1742u)$$

This second Δ*v* is faster because the reaction mass when you toss the second mass separately is a little bit less, so the acceleration of the flatcar during the throw is a little bit more.

Slide 9

Throw two masses, one at a time: $v_f = \frac{m}{M+2m}u + \frac{m}{M+m}u$

By induction . . .

- Throw three masses off, one at a time:

$$\Delta v = \frac{m}{M+3m}u + \frac{m}{M+2m}u + \frac{m}{M+m}u$$

- Throwing off four masses one at a time:

$$\Delta v = \frac{m}{M+4m}u + \frac{m}{M+3m}u + \frac{m}{M+2m}u + \frac{m}{M+m}u$$

- Throwing off *p* masses off, one at a time:

$$\Delta v = u\sum_{i=1}^{p} \frac{m}{M+im}$$

EXERCISE (to do in your notebook):

Suppose that you have a flatcar of mass 10*m*, and on it are four separate masses *m*. You can throw masses at a speed *u* relative to the flatcar.

(a) Determine the final speed of the cart if you throw them all at once.

(b) Determine the final speed of the cart if you throw them off one at a time.

Look at our expression for Δ*v*. For throwing two masses, one at a time, we get that v final is little m over big + plus two m times u, plus little m over big M plus one m times u. Notice that when we throw off two identical masses, one at a time we get two terms in the sum. The denominator of the first term is Big M plus two little m, and the denominator in the second term is big M + one little m

We get by induction that if we were to throw three identical masses, each of mass *m*, the result for delta v of the flatcar would have three terms, and the denominators would be big M + three little m, big M plus two little m, and big M plus one little m.

And if we were to throw four identical masses little m from the flatcar, one at a time, we'd get and expression with four terms in it, and the denominators of those terms would be Big M plus four little m, big M + three little m, big M plus two little m, and big M plus one little m.

$$\Delta v = \frac{m}{M+3m}u + \frac{m}{M+2m}u + \frac{m}{M+m}u$$

And throwing off four masses would give

$$\Delta v = \frac{m}{M+4m}u + \frac{m}{M+3m}u + \frac{m}{M+2m}u + \frac{m}{M+m}u$$

So throwing a total of p identical masses little m from the flatcar, one at a time, would give us an expression with p terms in it, with the denominators going from one times little m up to p times little m.

That is all expressed very concisely in the summation expression below. The total change in velocity of the flat car would be u times the sum from I equals 1 to p of little m divided by big M plus i times little m.

$$\Delta v = u\sum_{i=1}^{p} \frac{m}{M+im}$$

Take a moment to convince yourself that this pattern makes sense, and that the summation makes sense too.

Click
Here's an EXERCISE to do in your notebook:
Suppose that you have a flatcar of mass 10*m*, and on it are four separate masses *m*, and you can throw masses off the cart with a speed *u* relative to the cart.

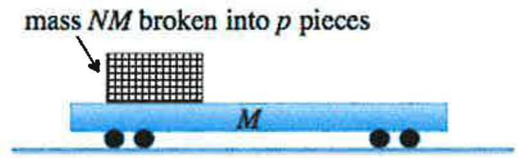
(a) Determine the final speed of the cart if you throw them all at once.
(b) Determine the final speed of the cart if you throw them off one at a time.

Slide 10

Toward real rockets...

We had derived

$$\Delta v = u\sum_{i=1}^{p}\frac{m}{M+im}$$



Generalize this:

- **Express total mass on the cart to be thrown off as some multiple *N* of the mass of the flatcar:**
  - Total mass being thrown off = *NM*
- **And, imagine that we break this mass up into *p* pieces and throw them off, one at a time.**
- **So the mass *m* being thrown off each time is**

$$m = \frac{\text{Total mass to be thrown off}}{\text{number of pieces we break it into}} = \frac{NM}{p}$$

Why do this?

→ Puts the expression for the final speed into a form that is easy to program a computer to evaluate!

Greetings! Welcome back

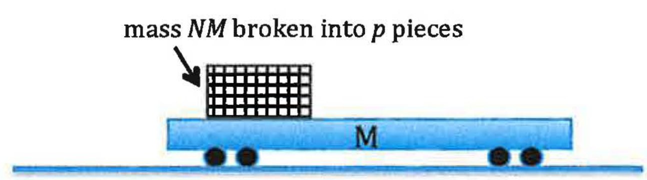
We had derived an expression for the change in velocity of the flatcar of mass big M when we threw p masses, each of mass little m, from the flatcar with a velocity u relative to the car.

Now we're going to make our previous expression for Δv can be made more general.

We can imagine that the total mass on the cart to be thrown off is some multiple *N* of the mass of the flatcar: Total mass being thrown off = *N times M*

So if, say, the mass on the fltacar were twice as much as the mass of the car itself, N would be two, and if it were eight times as much as the mass of the flatcart itself, N would be eight.

Click



And we can imagine that we break this mass up into *p* pieces and throw them off, one at a time.

In the example you did for the Exercise, *N* equaled 0.4 (you threw total mass 4 times little m off of a flatcar of mass 10 times little m) and *p* equaled four, since you broke the thrown mass into four equal pieces.

Click

So the mass *m* being thrown off each time is
The total mass being thrown off divided by the number of pieces you break it into, or N times capital M divided by p.

$$\left(m = \frac{\text{Total mass to be thrown off}}{\text{number of pieces we break it into}} = \frac{NM}{p} \right)$$

Click

Why do this, you might ask?
Because it lets us put the expression for the final speed into a form that is easy to program into a computer to evaluate, so we can look at breaking something into say, one thousand or one million pieces without having to do the calculation by hand.

Slide 11

Our expression for the final speed of the flatcar becomes:

$$\Delta v = u\sum_{i=1}^{p}\frac{m}{M+im} = u\sum_{i=1}^{p}\frac{\frac{NM}{p}}{M+i\frac{NM}{p}} = u\sum_{i=1}^{p}\frac{M\frac{N}{p}}{M\left(1+i\frac{N}{p}\right)}$$

$$= u\sum_{i=1}^{p}\frac{\frac{p}{N}}{\frac{p}{N}}\cdot\frac{\frac{N}{p}}{\left(1+i\frac{N}{p}\right)} = u\sum_{i=1}^{p}\frac{1}{\left(\frac{p}{N}+i\right)}$$

Here is some Python code that does this sum:

```
def in_pieces (times_M, how_many):
    '''This function takes as inputs how much mass is on a flat car as a multiple of the mass of the car itself ("times_M")
       and how many pieces you are breaking it into ("how_many") before you toss it off, one piece at a time.
       The function returns the change in speed of the car in as a multiple of the v_rel of the tossed mass.'''
    sum = 0
    for i in range(1,how_many+1):
        sum =sum+1/((how_many*1.0/times_M)+i)
    return sum
```

Here are the results if the total mass you are throwing off is equal to the mass of the flatcar itself:

Number of pieces you break the mass into	Δv (as a multiple of $u = v_{rel}$)
1	0.5
2	0.583333333333
3	0.616666666667
4	0.634523809524
5	0.645634920635
10	0.668771403175
20	0.680803381793
50	0.68817217931
100	0.690653430482
1000	0.69289724306
1000000	0.69314693056

Equals:

$$\ln(2) = \ln\left(\frac{\text{initial mass of the cart + tossed mass}}{\text{final mass of the empty cart}}\right)$$

Not **a coincidence!**

Our expression for the final speed of the flatcar starts out where we were before,
Substitutes N time capital M over p for little m, and then does some algebra to get to the final form of the expression.

$$\Delta v = u\sum_{i=1}^{p}\frac{m}{M+im} = u\sum_{i=1}^{p}\frac{\frac{NM}{p}}{M+i\frac{NM}{p}} = u\sum_{i=1}^{p}\frac{\cancel{M}\frac{N}{p}}{\cancel{M}\left(1+i\frac{N}{p}\right)}$$

$$= u\sum_{i=1}^{p}\frac{\frac{p}{N}}{\frac{p}{N}}\cdot\frac{\frac{N}{p}}{\left(1+i\frac{N}{p}\right)} = u\sum_{i=1}^{p}\frac{1}{\left(\frac{p}{N}+i\right)}$$

The advantage of this form of the equation is that we can input just two variables: What is the mass being thrown off relative to the mass of the empty flatcar, and how many pieces are we breaking this mass up into? Our answer for the change in velocity of the cart will come out to be some number multiplied by the relative velocity of the thrown mass.

Click

Here is all of the Python code it takes to do the calculation.

In this code, '"times_M" is our capital N (how big the thrown mass is compared to the mass of the cart) and "how_many" is our p, *how many pieces we're breaking the thrown mass up into.*

```
def in_pieces (times_M, how_many):
    '''This function takes as inputs how much mass is on a flat car as a multiple of the mass of the car itself ("times_M")
       and how many pieces you are breaking it into ("how_many") before you toss it off, one piece at a time.
       The function returns the change in speed of the car in as a multiple of the v_rel of the tossed mass.'''
    sum = 0
    for i in range(1,how_many+1):
        sum =sum+1/((how_many*1.0/times_M)+i)
    return sum
```

Click

Here are the results if the total mass you are throwing off is equal to the mass of the flatcar itself (N = 1).

Number of pieces you break the mass into	$\Delta v_{flatcar}$ (as a multiple of $u = v_{rel}$)
1	0.5
2	0.583333333333
3	0.616666666667
4	0.634523809524
5	0.645634920635
10	0.668771403175
20	0.680803381793
50	0.68817217931
100	0.690653430482
1000	0.69289724306
1000000	0.69314693056

Click

It is interesting to note that these results converge, and that this last result is identical to ln(2) , where

$$\ln(2) = \ln\left(\frac{\text{initial mass of the cart} + \text{tossed mass}}{\text{final mass of the empty cart}}\right)$$

This is not a coincidence! How exciting to find this result just pop out from our calculation! What is going on?

Slide 12

Look at the expression $\Delta v = u\sum_{i=1}^{p} \frac{m}{M+im}$

- If the number of pieces we are breaking things up into is very large, then the amount of mass *m* being thrown off each time is small, so we can call each piece ***dm***.
- The quantity in the denominator is ranging, in increments of *dm*, from the final mass *M* (the empty flatcar by itself) to *M* + *pm*, the initial mass of the car with all of its load.

If thrown mass = flatcar mass
N = 1 → ln 2

The sum becomes:

$$\Delta v = u\sum_{i=1}^{p}\frac{m}{M+im} = u\int_{M_f}^{M_0}\frac{dm}{m} = u\ln\left(\frac{m_{\text{final}}}{m_0}\right) = u\ln\left(\frac{(N+1)M}{M}\right) = u\ln(N+1),$$

Let's look at our expression for the change in the velocity of the flatcar

$$\Delta v = u\sum_{i=1}^{p}\frac{m}{M+im}$$

If the number of pieces we are breaking things up into is very large, then the amount of mass *m* being thrown off each time is small, so we can call it *dm*.

Click

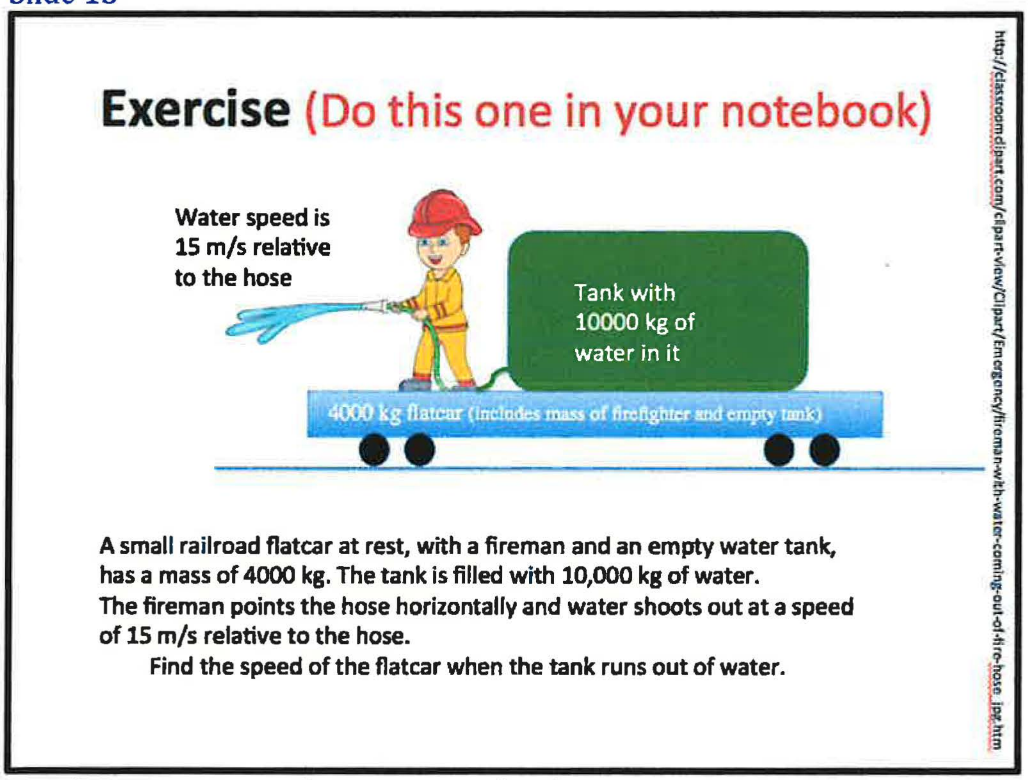
And the quantity in the denominator is ranging, in increments of *dm*, from the final mass *M* (the empty flatcar by itself) to big *M* + *p times m*, the mass of the flatcar fully loaded. (soon we'll make the analogy that big M is the mass of the empty rocket, and that p times little m is the mass of all of the fuel and oxidizer, and that the exhaust speed will be u, and if we look at throwing the mass out continuously, dm is the amount of fuel thrown out in a time dt.)

Click

Click
The m in the sum becomes dm in the integral, and *M plus* im becomes just the mass of the system over the time that system mass is changing.
The sum becomes *u* times the integral of dm/m, where m ranges from the final mass of the system (just the empty flatcar by itself) to the initial mass of the system (flatcar plus all of the mass on it, which is capital *M* plus *N times* capital *M, or N plus one times M,* where *N* represented how large the mass being tossed off was as a multiple of the mass of the flatcar by itself. The *M's divide out, and we're left with u times log (N + 1).*
In the example we worked through in the table, we broke up the mass into a million pieces, N was equal to one and our result was that the final speed of the flatcar was u times log 2.
How nice that the calculus works out and gives us the same result as we got by treating the tossed mass as 1 million individual discrete bits tossed off one at a time!

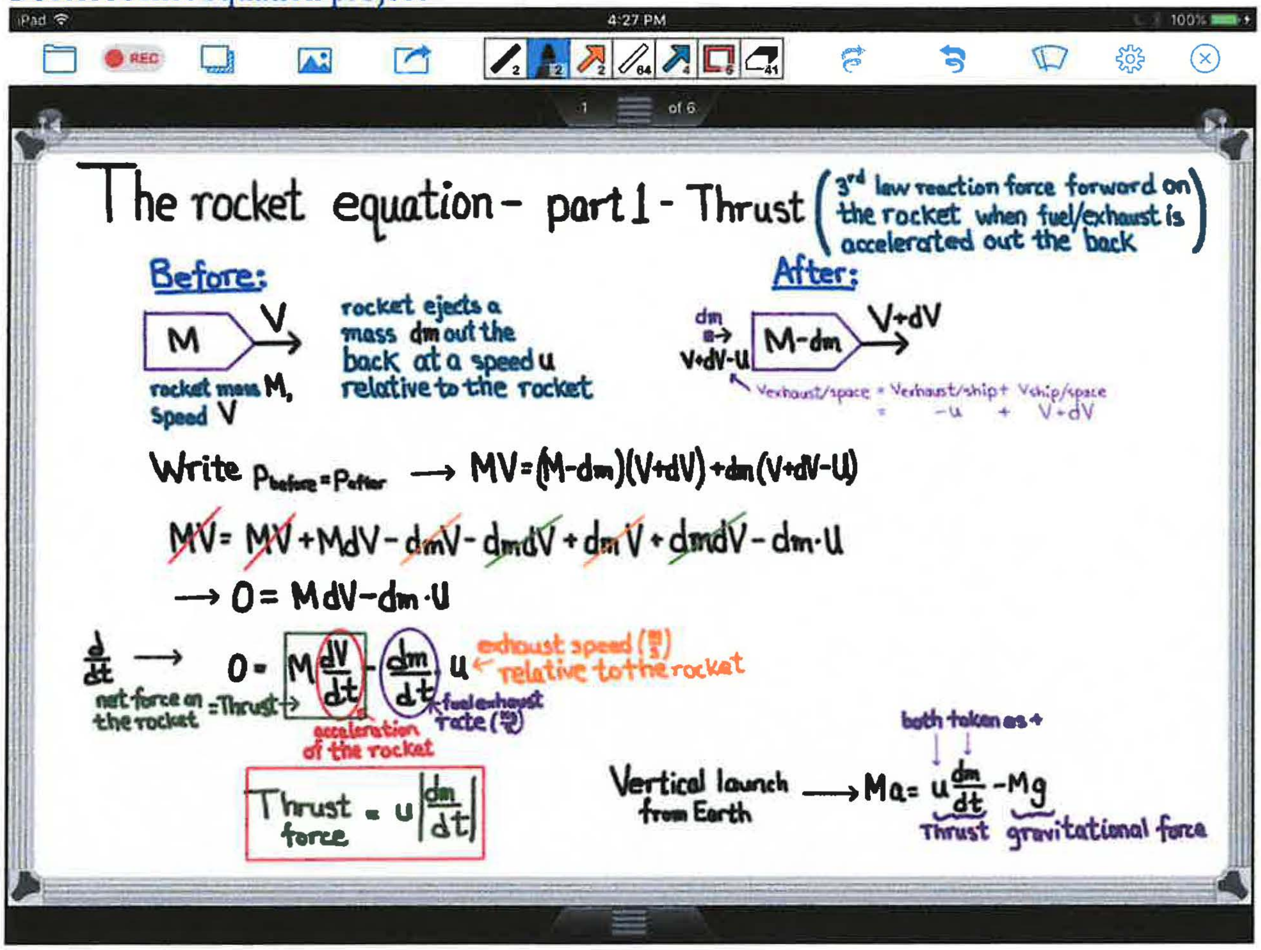
$$\Delta v = u\sum_{i=1}^{P}\frac{m}{M+im} = u\int_{M_f}^{M_0}\frac{dm}{m} = u\ln\left(\frac{m_0}{m_f}\right) = u\ln\left(\frac{(N+1)M}{M}\right) = u\ln(N+1),$$

Slide 13



Here's an exercise to do in your notebook.
A small railroad flatcar at rest, with a fireman and an empty water tank, has a mass of 4000 kg. The tank is filled with 10,000 kg of water.
The fireman points the hose horizontally and water shoots out at a speed of 15 m/s relative to the hose. Find the speed of the flatcar when the tank runs out of water.

Script for the Rocket Equation using calculus
Doceri rocket equation project



Click to first stop

Greetings! Welcome back.

There are two rocket equations we'll derive. The first one has to deal with thrust, or the forward force on a rocket when fuel or exhaust is accelerated out the back.

The idea for a rocket is that we continuously eject mass out the back to provide a reaction force forward.

Click

We consider a rocket of mass capital *M,* moving in the positive direction at speed *V.*

The rocket ejects a mass *dm* of propellant or exhaust in the negative direction with a velocity minus *u* relative to the ship.

Note that *dm* and *u* are taken to be positive quantities.

Click

Using conservation of momentum, before the ejection of mass *dm* the momentum of the ship system was *MV.*

After the ejection, the ship has a mass *M-dm* and a velocity V + dV, and the exhaust has a mass *dm* and a velocity of *V + dV –u.*

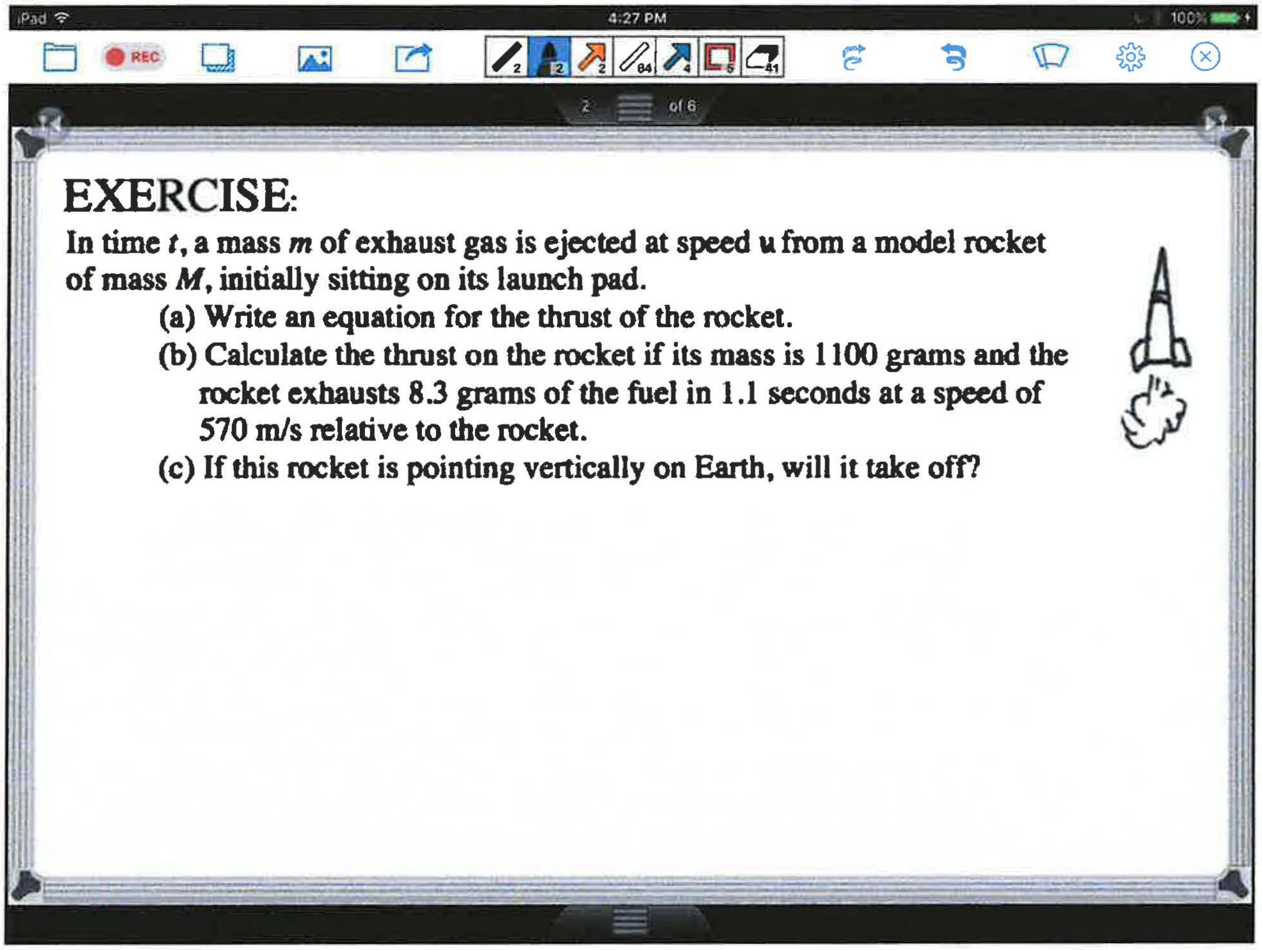
Click
(It's v+dv because that is the new velocity of the ship when the exhaust finally leaves the ship.)

Speed 8 Click
We can write p before equals p after, multiply things out, and eliminate things that either are on both sides of the equation, or which appear with plus and minus signs on the same side of the equation.
Click
This gives us zero equals M dv – dm times u.
Click
We can take the time derivative of both sides, which gives M dv/dt minus u dm/dt equals zero.
Speed 3 Click
Let 's take these terms one at a time:
Dv/dt is the acceleration of the rocket.
and
Dm/dt is the rate at which exhaust is being ejected from the rocket, in units of kilograms per second.
Speed 8 Click
And u is the speed of the exhaust relative to the ship.

Click
M dv/dt is Ma, which is an expression for the net force on the ship.

Rearranging gives that the thrust, which is the net force on the ship, is equal to u dm/dt—the relative velocity (in m/s) times the mass ejection rate (kg/s).
Click
If this is a vertical launch from earth, we have to include a gravity term in the net force, where now the mass is a function of time.

Slide 2



Here's an EXERCISE to try: Give it a shot. I'll see you again in a few minutes.

In time *t*, a mass *m* of exhaust gas is ejected at speed *u* from a model rocket of mass *M*, initially sitting on its launch pad.

(a) Write an equation for the thrust of the rocket.

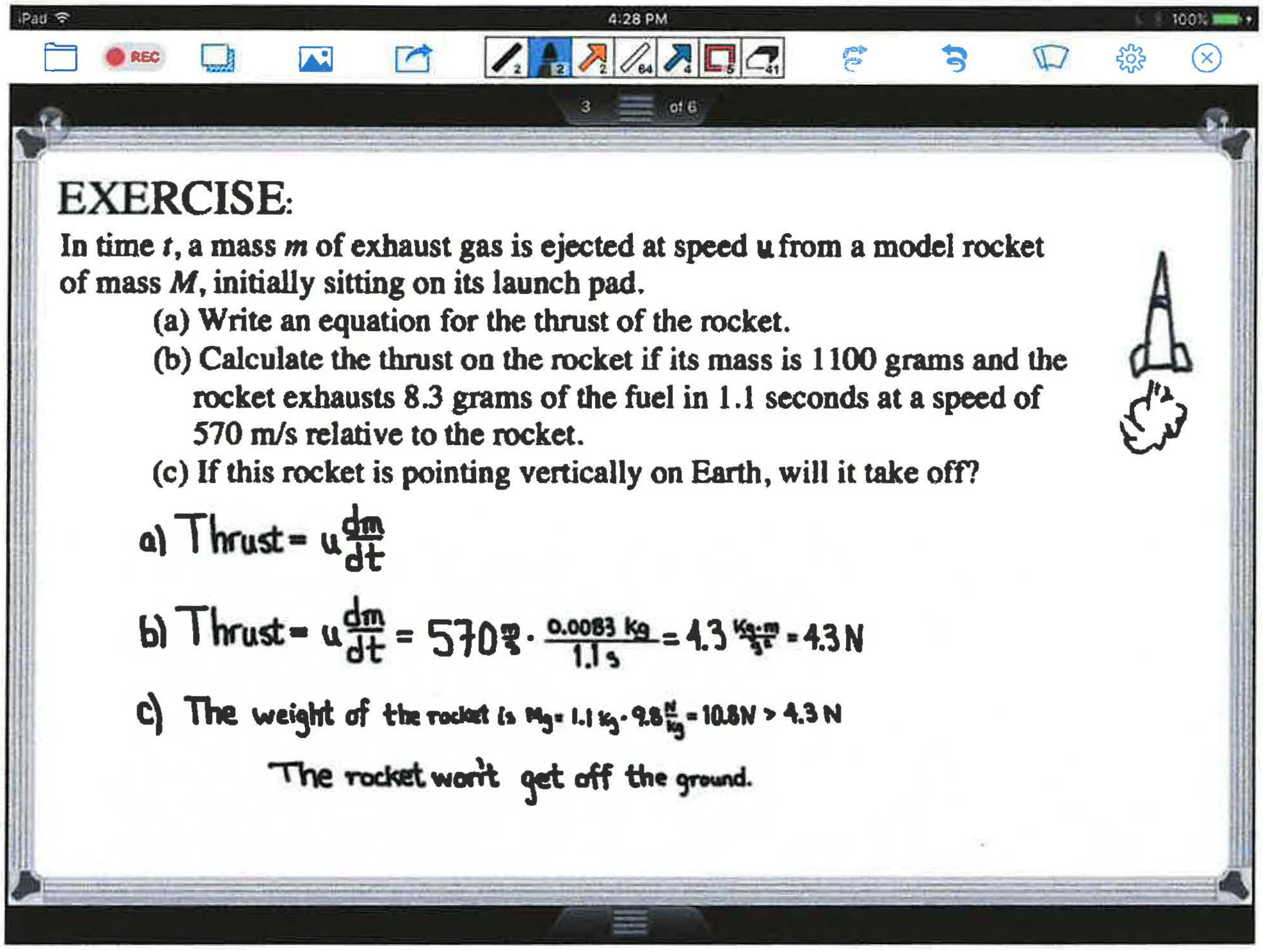
(b) Calculate the thrust on the rocket if its mass is 1100 grams and the rocket exhausts 8.3 grams of the fuel in 1.1 seconds at a speed of 570 m/s relative

to the rocket.

(c) If this rocket is pointing vertically on Earth, will it take off?

Begin new segment

Slide 3



We were looking at a problem involving a rocket of mass capital *M*, ejecting mass little *m* in a time *t* at a speed *u* relative to the rocket.

Speed 8 Click

Part (a) asks us to write an equation for the thrust of the rocket. That's just u times dm/dt.

Click

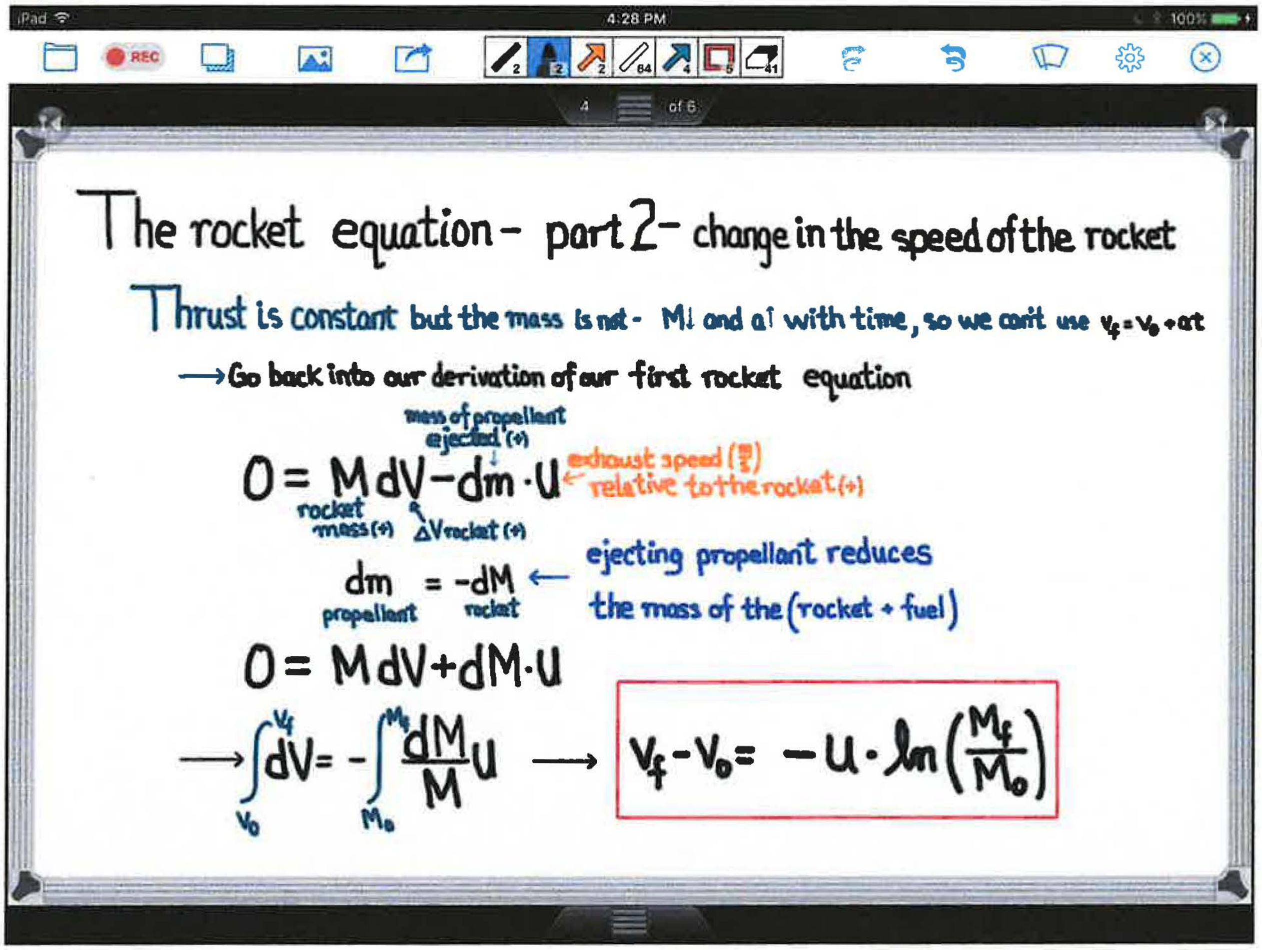
Part (b) gives us some numbers to plug in. In order to get a force in Newtons the masses have to be in kg.

So we plug in numbers, 570 m/s times 0.0083 kg divided by 1.1 s, to get a thrust of 4.3 N.

Click

Part (c) asks us to determine if the rocket gets off the ground. Since the weight of the rocket is over ten newtons and the thrust is only 4.3 N, the rocket doesn't ever take off. Waah.

Slide 4



Speed 8 Click

Now we'll look at a second rocket equation, which lets us relate the change in speed of the rocket to the mass and exhaust speed of the fuel.

Our assumption is that the thrust is constant but that the mass of the rocket decreases with time, so the acceleration of the rocket *increases* with time, so we can't use our constant acceleration kinematics equations here.

Speed 10 Click

We go partway back into our derivation of our first rocket equation, for thrust, and pick up at the point where we had

Zero equals M dV – dm times u.

U is taken as a speed, so a positive number, and dm is taken as a positive number too, because it is the amount of mass being exhausted from the rocket.

Click

The next thing we do is recognize that when a positive amount of propellant is thrown out the back of the rocket, the change in the mass of the *rocket* is *negative*. So we will write that d little m equals *minus* d capital M

Click

Now we rewrite the previous equation as zero equals capital *M* dV plus d capital M times u. Now all of the masses in the equation refer to the mass of the rocket.

Click

We can rearrange things a bit to get dV equals – dM over M times u.

Click

Now we integrate dv from v initial to v final, and dm/M from the initial mass to the final mass.

Click

This gives that the change in speed of the rocket equals – U times the log of the final mass divided by the initial mass.

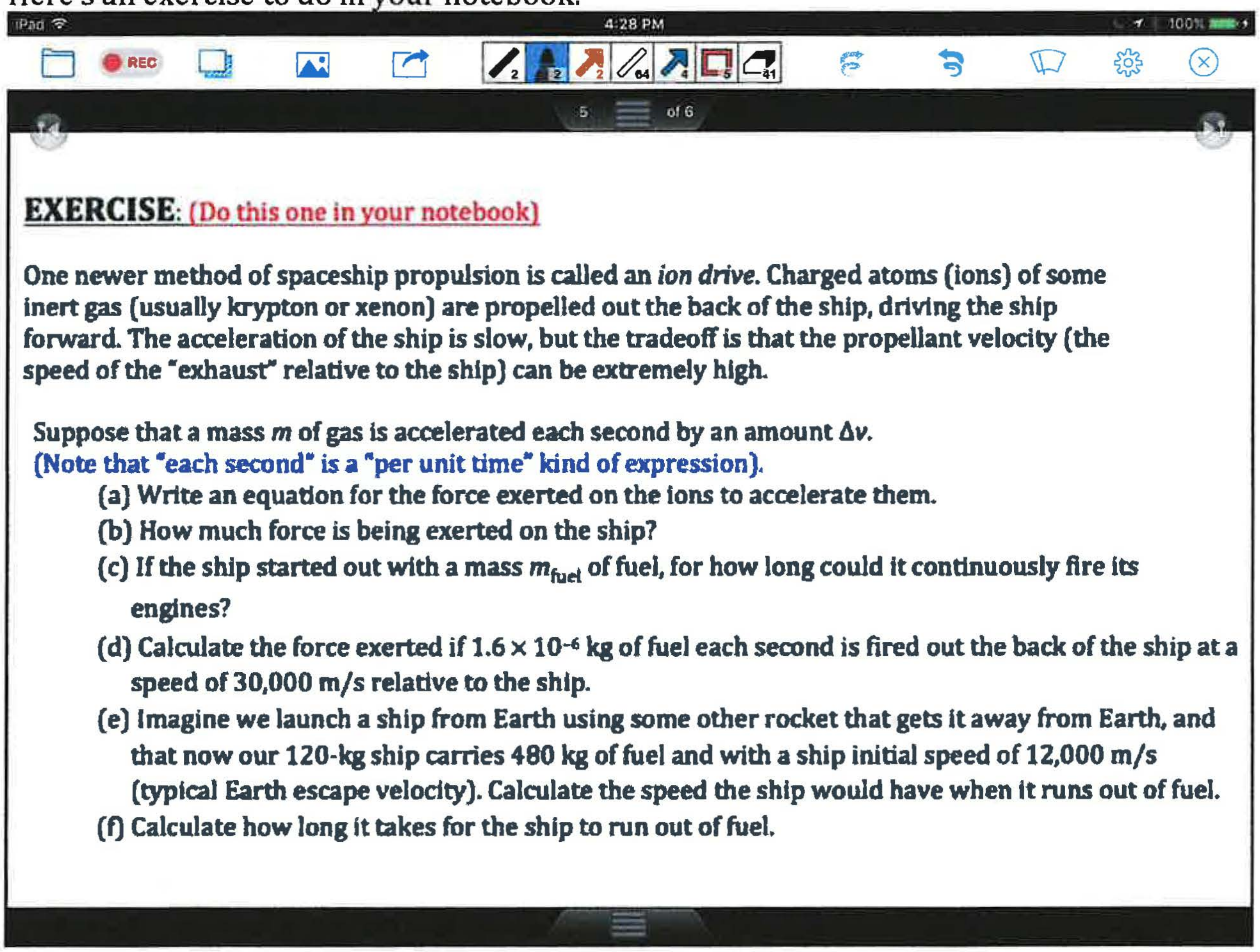
(the minus sign makes sense there because the ratio of the final to initial mass is less than one, so the log is negative).

This is the same answer we got by considering the limit of the discrete case, where we threw the mass off just a little bit at a time. Hooray for calculus!

Click

Slide 5

Here's an exercise to do in your notebook:



Using Calculus to Find

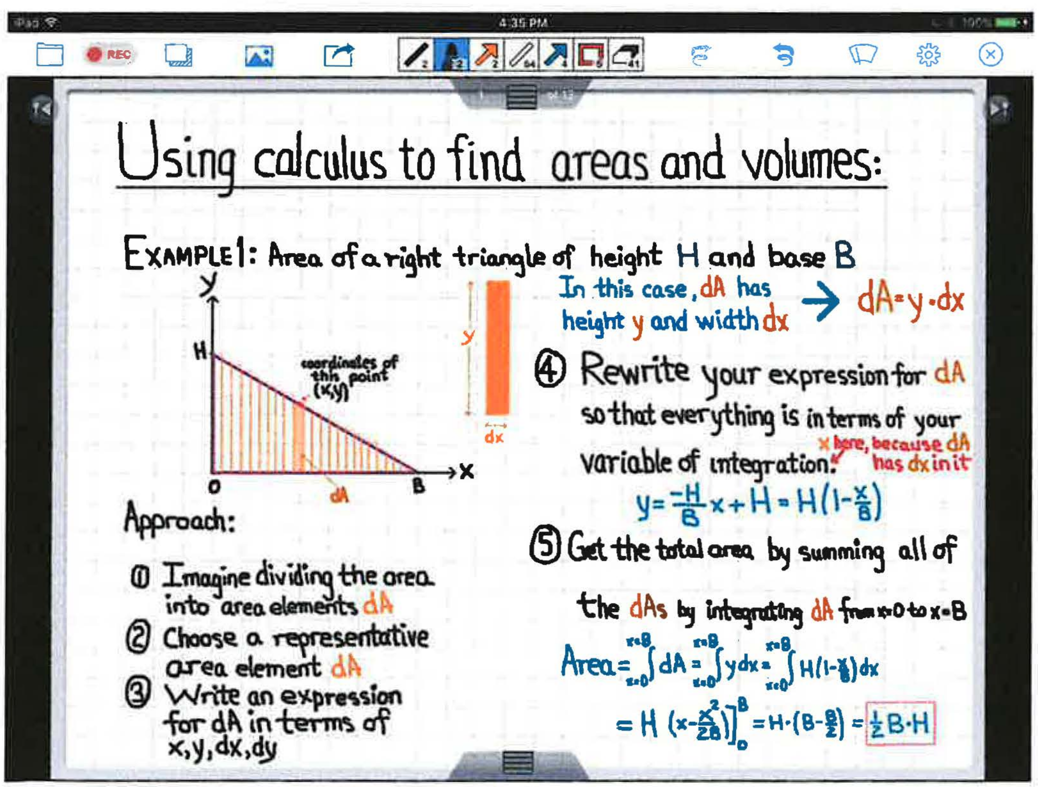
Areas & Volumes

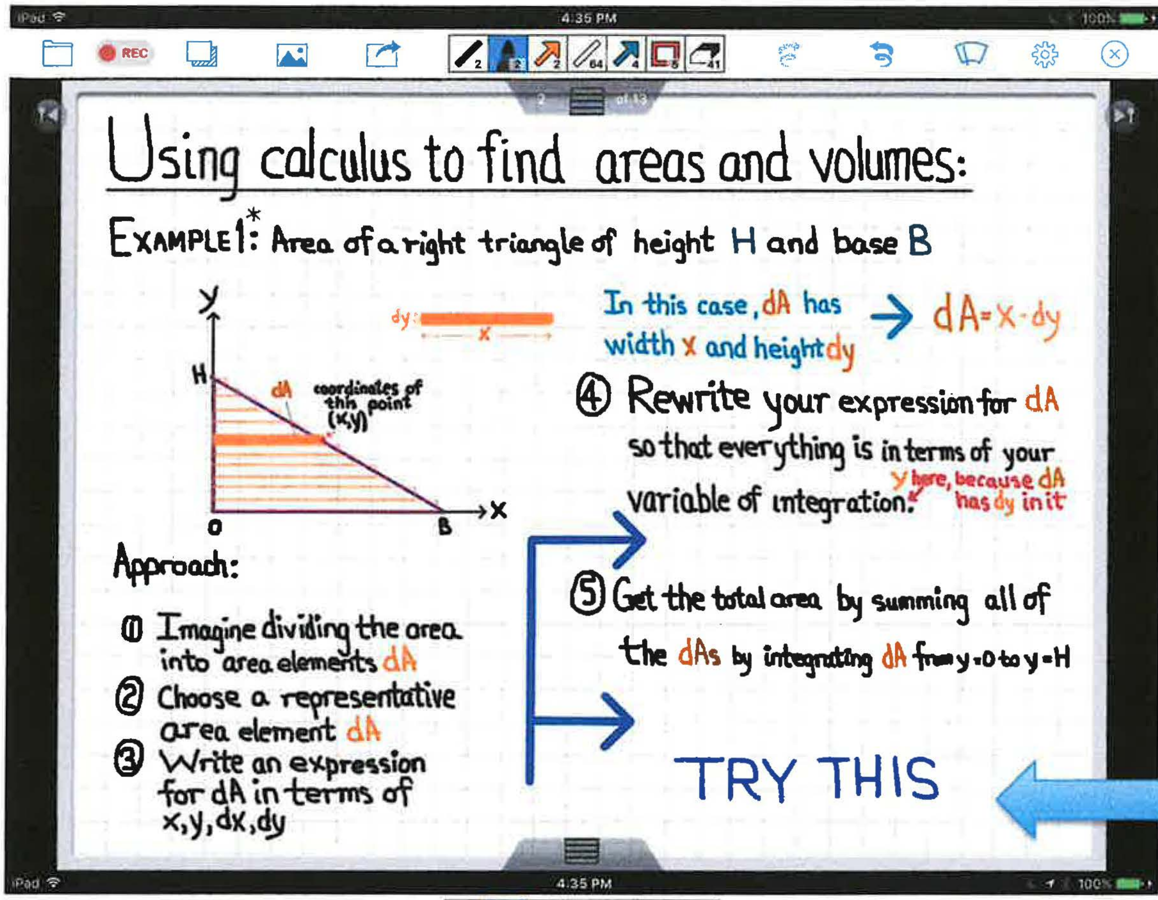
Using Calculus to find Areas and Volumes:

This one is not yet scripted. Because it is all derivations, the script will look remarkably like talking through the process of finding areas and volumes using calculus by pretty much reading/narrating the slides as the text and drawings appear.

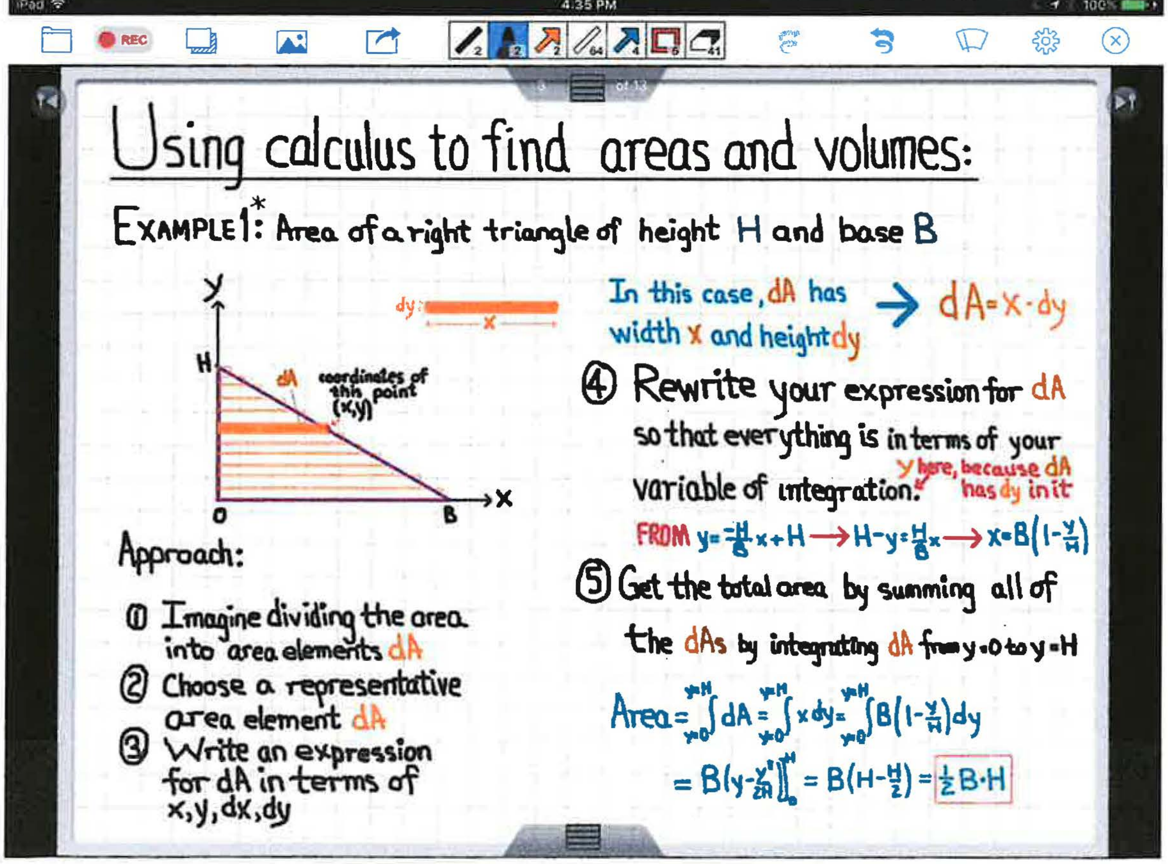
All of this is done in Doceri. The first parts of the slide (title, initial diagram) will appear all at one. Each step after that will appear on the screen in the lecture video as though I was writing, quickly and neatly, on the board as fast as I could say things.

Many Physics 4A students struggle with this material, which is presented somewhat it Math 181 and also in Math 280. For material like this is should be especially beneficial to students to be able to stop things mid-derivation or to rewatch them. It was in part a student question a couple of years ago—"Isn't there some video on this that I can watch?"—that was part of my original motivation for this sabbatical.



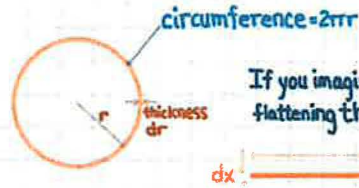


This would be the end of the first video, where students try to finish the problem on their own, with the solution given in the next video.



EXAMPLE 2: Area of a circle of radius R

Each dA is a thin ring of thickness dr and (variable) radius r



If you imagine cutting and then flattening the ring:

$dA_{ring} = 2\pi r \cdot dr$

① Divide into area elements dA

② Choose a representative area element dA

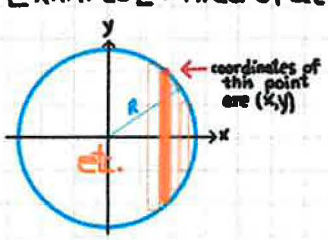
③ Write an expression for dA in terms of r, dr

④ Everything in dA is already in terms of a single variable r.

⑤ Get the total area by summing all of the dAs by integrating dA from r=0 to r=R

$$\text{Area} = \int_{r=0}^{r=R} dA = \int_{r=0}^{r=R} 2\pi r \cdot dr = 2\pi \cdot \frac{r^2}{2}\Big]_0^R = \boxed{\pi R^2}$$

EXAMPLE 2*: Area of a circle of radius R



$dA = 2y \cdot dx$

Equation for the circle is $x^2 + y^2 = R^2$

$\rightarrow y = \sqrt{R^2 - x^2} \rightarrow dA = 2\sqrt{R^2 - x^2}\, dx$

① Divide into area elements dA

② Choose a representative area element dA

③ Write an expression for dA in terms of x, y, dx, dy

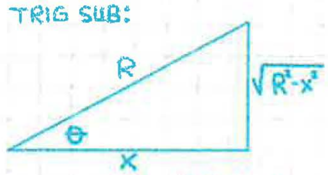
④ Rewrite your expression for dA so that everything is in terms of your variable of integration: x here, because dA has dx in it

⑤ Get the total area by summing all of the dAs by integrating dA from x=−R to x=+R

$$\text{Area} = \int_{x=-R}^{x=+R} dA = \int_{x=-R}^{x=+R} 2\sqrt{R^2 - x^2}\, dx$$

$$\text{Area} = \int_{x=-R}^{x=R} dA = \int_{x=-R}^{x=R} 2\sqrt{R^2-x^2}\,dx$$

TRIG SUB:



$x = R\cos\theta$: $\theta = 0 \leftrightarrow x = +R$, $\theta = \pi \leftrightarrow x = -R$

$dx = -R\sin\theta\, d\theta$

From the triangle: $\sqrt{R^2-x^2} = R\sin\theta$

$$\text{Area} = \int_{\theta=\pi}^{\theta=0} 2\underbrace{R\sin\theta}_{\sqrt{R^2-x^2}} \cdot \underbrace{(-R\sin\theta\, d\theta)}_{dx}$$

identity for $\sin^2\theta$

$$= -R^2\int_\pi^0 2\sin^2\theta\, d\theta = -R^2\int_\pi^0 2\left(\frac{1-\cos 2\theta}{2}\right)d\theta$$

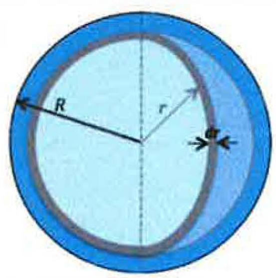
$$= -R^2\int_\pi^0 d\theta + \frac{R^2}{2}\int_\pi^0 \cos 2\theta\; 2d\theta$$

looks like $\cos u\, du$ where $u = 2\theta$

$$= -R^2\,\theta\Big]_\pi^0 + \frac{R^2}{2}\sin 2\theta\Big]_\pi^0$$

$$= -R^2(0-\pi) + \frac{R^2}{2}(0-0) = \boxed{\pi R^2}$$

Example 3: Volume of a sphere of radius R



Approach:

① Divide into volume element dV

Thin spherical shells of (variable) radius r and thickness dr

② Choose a representative volume element dV (in gray, on the diagram)

③ Write an expression for dV in terms of your variables (r, dr)

Surface area of a sphere $= 4\pi r^2$ (this is the area it would have if you could slice the shell and lay it out flat)

Thickness of the spherical shell $= dr$

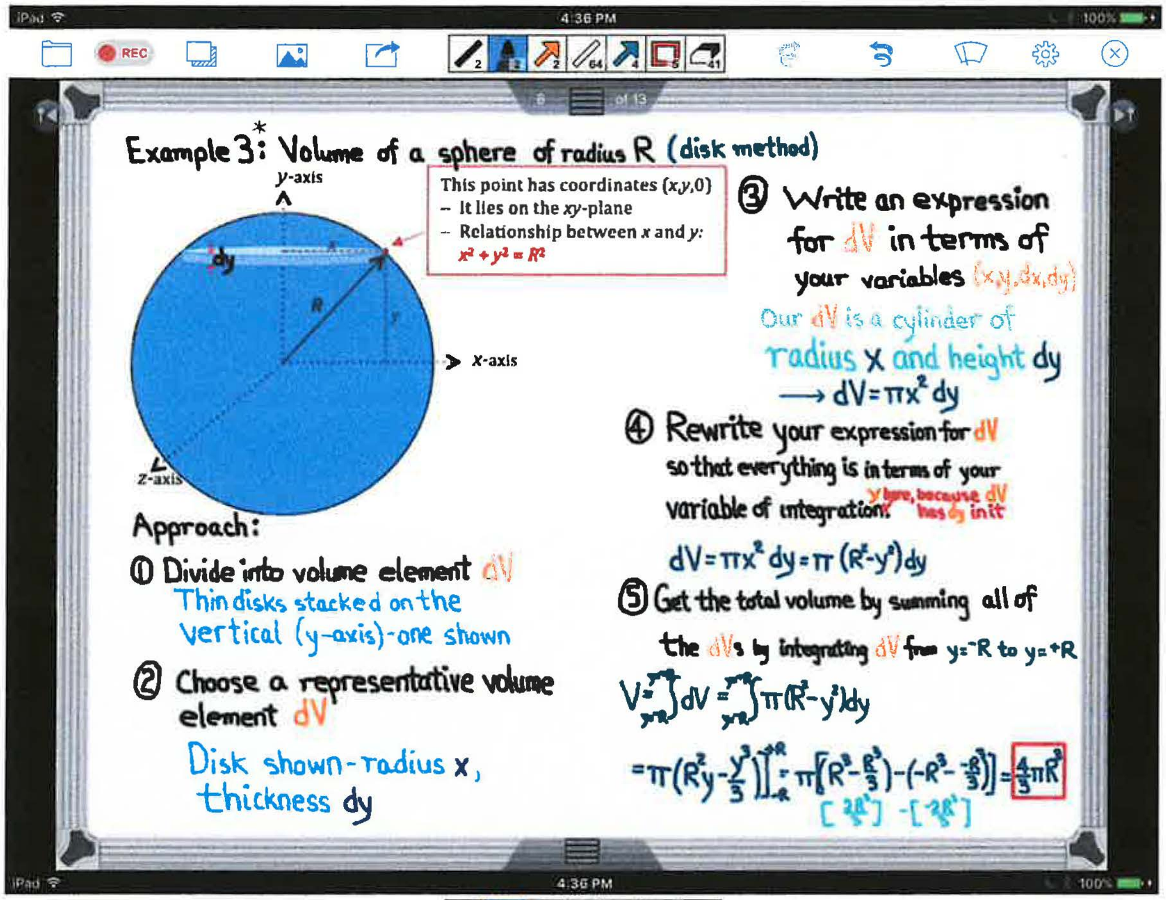
$\rightarrow dV = 4\pi r^2 dr$

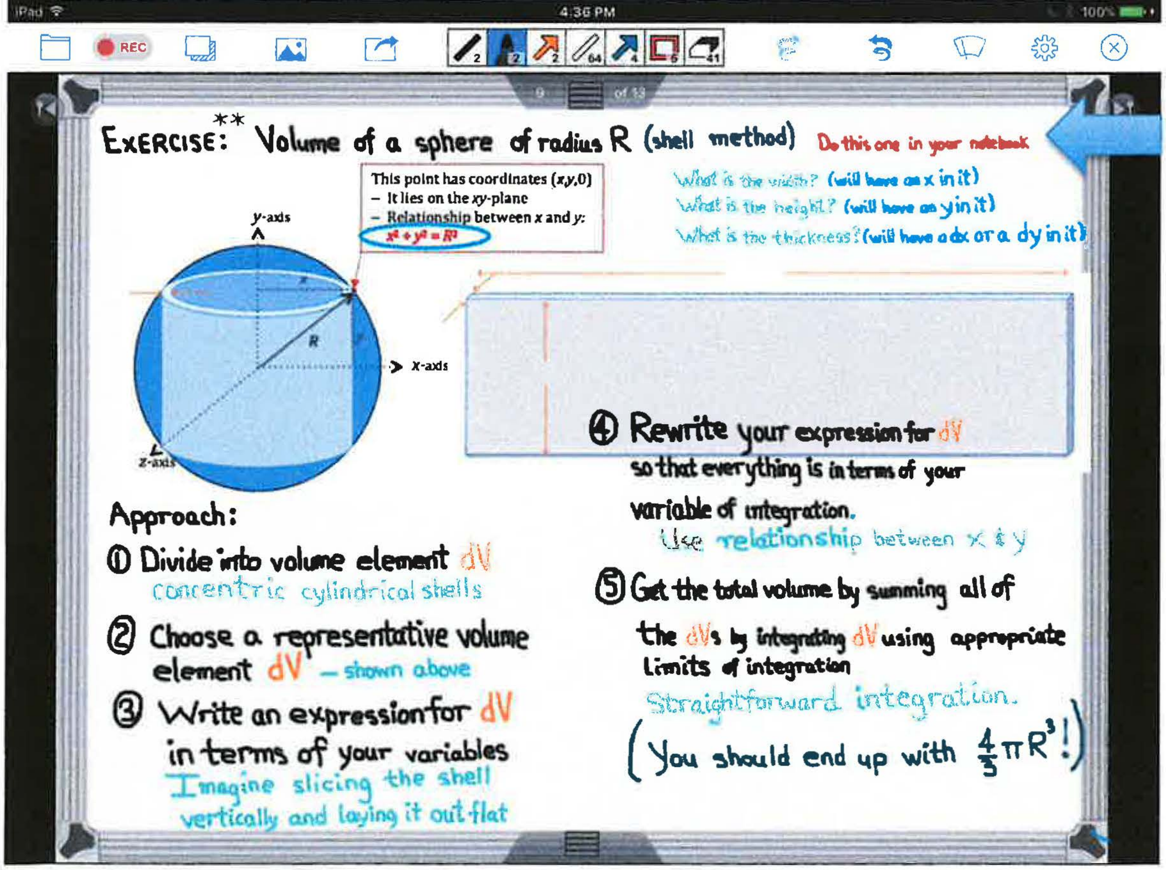
④ Everything in dV is already in terms of a single variable of integration, r.

⑤ Get the total volume by summing all of the dVs by integrating dV from r=0 to r=R.

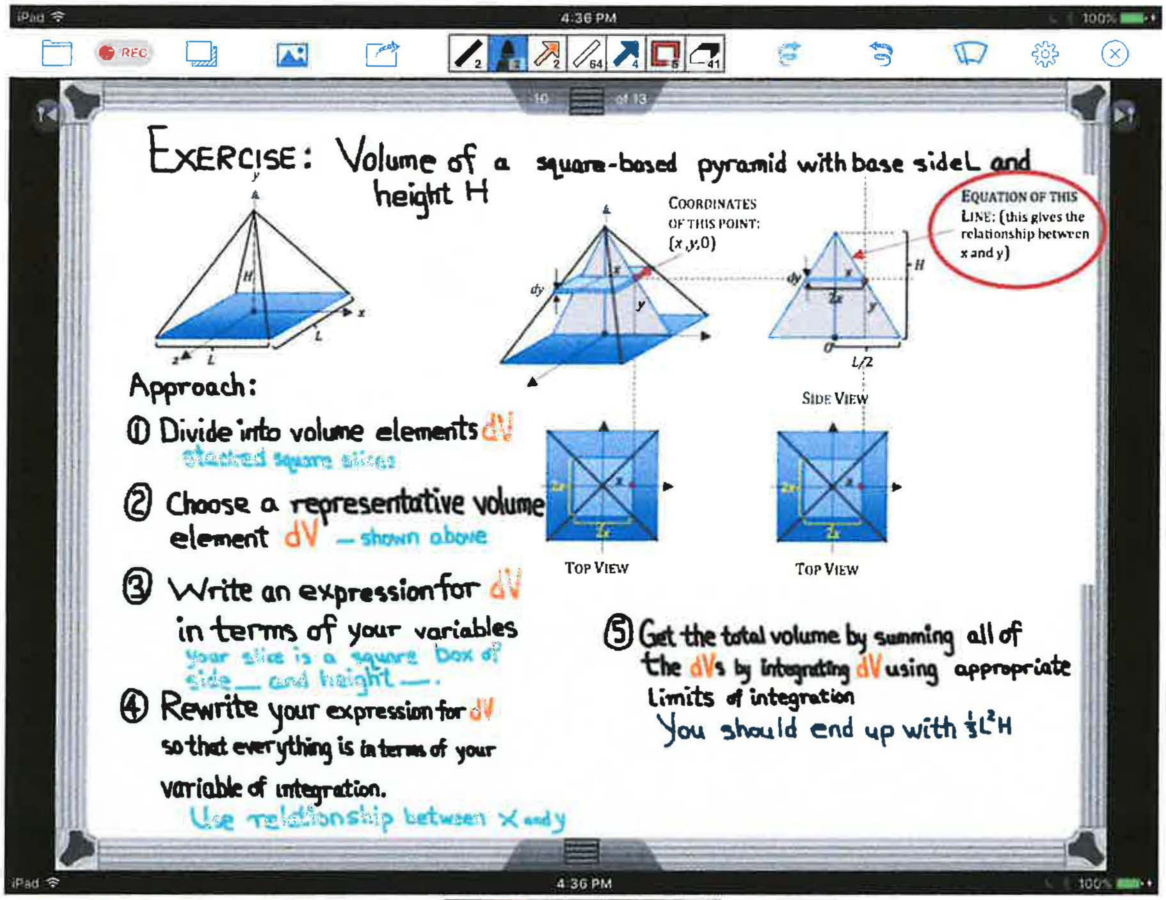
$$V = \int_{r=0}^{r=R} dV = \int_{r=0}^{r=R} 4\pi r^2 dr = 4\pi \frac{r^3}{3}\Big]_0^R$$

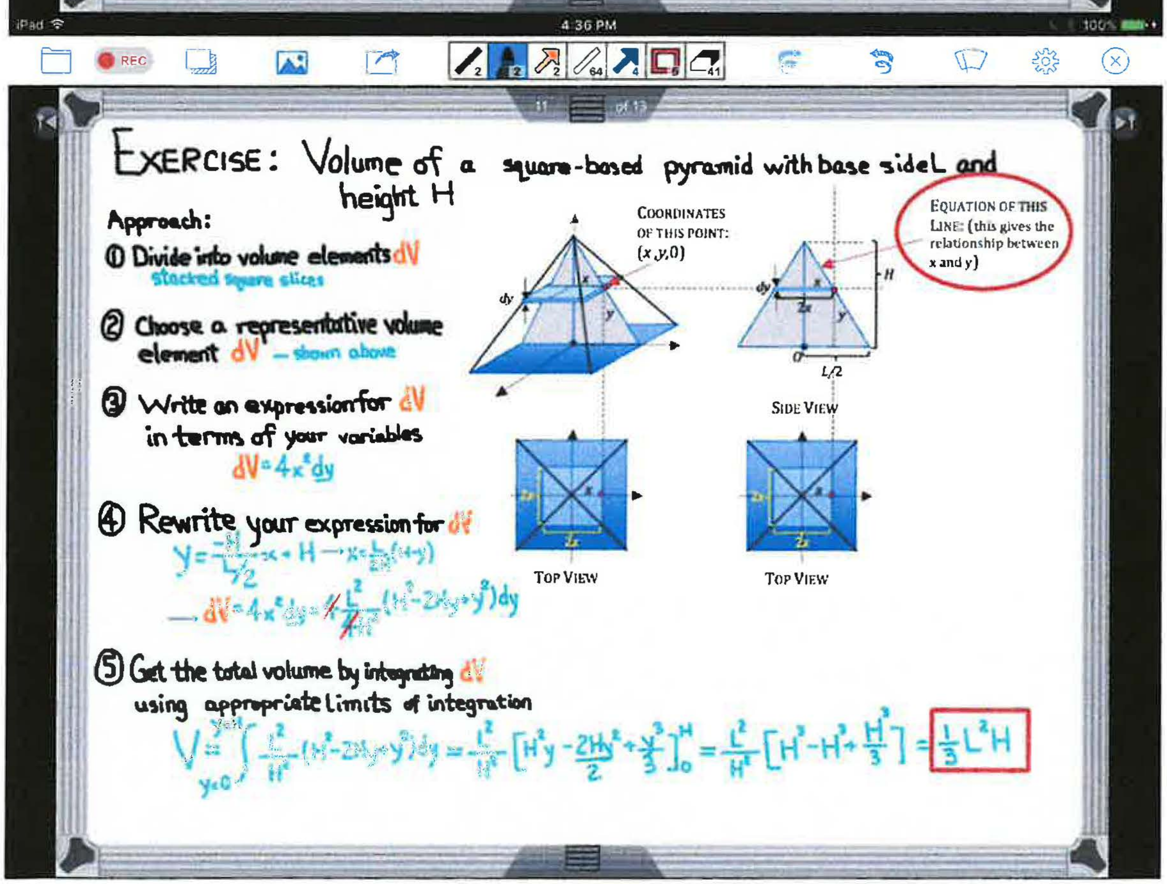
$$= \boxed{\frac{4}{3}\pi R^3}$$

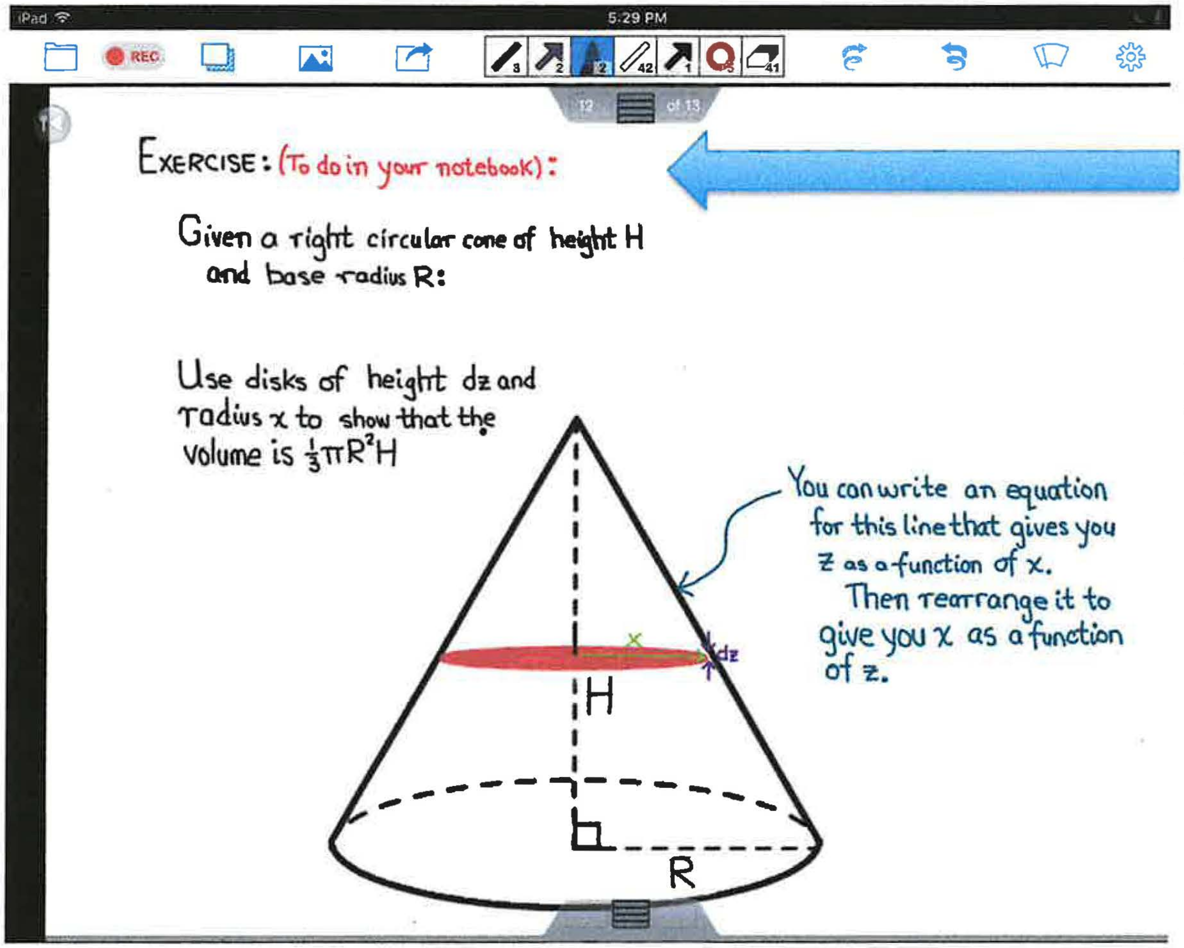




This would be the end of the next video, where students have an exercise to do in their notebooks to turn in. No solution is given in the video.


EXERCISE: Volume of a square-based pyramid with base side L and height H
COORDINATES OF THIS POINT: (x,y,0)
EQUATION OF THIS LINE: (this gives the relationship between x and y)
SIDE VIEW
TOP VIEW
TOP VIEW
Approach:
① Divide into volume elements dV
stacked square slices
② Choose a representative volume element dV — shown above
③ Write an expression for dV in terms of your variables
your slice is a square box of side __ and height __.
④ Rewrite your expression for dV so that everything is in terms of your variable of integration.
Use relationship between x and y
⑤ Get the total volume by summing all of the dVs by integrating dV using appropriate limits of integration
You should end up with ⅓L²H


EXERCISE: Volume of a square-based pyramid with base side L and height H
COORDINATES OF THIS POINT: (x,y,0)
EQUATION OF THIS LINE: (this gives the relationship between x and y)
SIDE VIEW
TOP VIEW
TOP VIEW
Approach:
① Divide into volume elements dV
stacked square slices
② Choose a representative volume element dV — shown above
③ Write an expression for dV in terms of your variables
dV = 4x²dy
④ Rewrite your expression for dV
⑤ Get the total volume by integrating dV using appropriate limits of integration
= ⅓L²H



This would be the end of the last video, where students have an exercise to do in their notebooks to turn in. No solution is given in the video.

VPython Videos—Roughly in the order that I made them.

Here is a link to the playlist, where all of these videos reside:
http://www.3cmediasolutions.org/f/da53d37be942a13836b0a8df96c9a705dceaabf8
(They are in alphabetical order in the playlist. The playlist is three "pages" long on the host server, so if you don't see a file there it is likely on the next or next-next page.)

Many of these programs went through several revisions. What is shown here is mostly the final revision of each program.

Ball_toss_straight_up.py (ball tossed from the ground, reaching max height, returning to the ground)

Ball_toss_at_angle.py (ball tossed from the ground at an angle, reaching a max height, striking the ground again)

Box_sliding_up_inclined_plane.py (challenge here was to draw the inclined plane. Learned how to do an *extrusion*, where you draw a two-dimensional shape (say, a triangle or a circle) and then extend it following some path into three-D (so, a triangle can follow a straight path to become a wedge, or a circle can follow a circular path to become a donut.

Mass_spring_on_inclined_plane.py (mass oscillating on a spring, parallel to an inclined plane)—challenge here was to animate the length of the spring to change with the movement of the mass along the spring

hockey_puck_on_a_table.py, then **Three_hockey_pucks**.py (three hockey pucks given identical initial impulses on surfaces with different coefficients of friction. Smoother surface = puck goes farther)—part of a model I use for teaching Newton's first law. Part of the challenge here was learning how to introduce a delay between different parts of the program. The standard command "sleep(4)" = stop for four seconds doesn't quite do that. I came up with a work around which involved inserting a loop that ran *very slowly* in between the first puck part of the program and the second part of the program.

Stopper.py (uses an extrusion to create the shape of a rubber stopper). Takes a circle, extrudes along a straight path but changes the scale of the radius of the circle over the length of the path

cart_spring_on_track.py (cart on a track, force sensor on the cart, stopper mounted on the force sensor. Cart goes along a track and the stopper runs into a spring, then rebounds back. Program animates the collision and shows the force vs. time, velocity vs. time, force vs. position, and kinetic energy vs. position in "real time" for the collision.

simple_pendulum.py (illustrates a mass on a string, oscillating back and forth, with arrows showing how the x and y locations of the mass change with time)

vertical_mass_spring.py
Illlustrates a lab where we have a motion sensor on the floor, a spring hanging from a support at some height, and a mass hanging from the spring. First we locate the position of the bottom of the mass when we support the mass so that the spring is unstretched, then the equilibrium position of the bottom of the mass when we slowly lower it and let it settle, and then the motion of the mass-spring system as we slightly raise the mass and then let it go. Graphs the height and velocity vs. time for the mass)

Meterstick_clay_collision.py
Illustrates a lab where a meter stick is pivoted at one end and has a nail on the other end. The meter stick is lifted so that it is horizontal, then let go. The meterstick swings into and sticks with a clump of clay at the bottom of the swing and then rises to some final position and swings back and forth.

centripetal_force_lab.py
Illustrates a lab where we have a rotating turntable with a force sensor with a stopper mounted on it, and shows how the direction of the force changes (but always points inward) as the turntable rotates.)

Spring_Constant3_IDLE.py
Illustrates a lab with two different hanging springs, and plots the stretch vs. hanging weight as progressively more weight is hung on each spring.

Spring_constant4.py
Horizontal mass spring system showing the system at equilibrium, then the mass is pulled to one side and released and starts oscillating. There are arrows that change length and location showing the force and displacement with time and labels that move with them.

rotary_motion.py
Shows an object traveling in a circle at constant speed. Shows the acceleration, velocity, and position vectors with labels that move and change direction as the object moves.

spring_array.py draws a square array of masses connected by springs

Friction_demo.py shows the motion of a block on a sticky horizontal surface as the pulling force slowly increases. Graphs the friction force, pulling force, and acceleration of the block as a function of time

banked_road.py A problem we solve involves a car making a turn on a banked road without friction. This illustrates the gravitational and normal forces acting on the car as it moves along a circular banked turn.

banked_road2.py A problem we solve involves a car making a turn on a banked road without friction. This illustrates the gravitational and normal forces acting on the car as it moves along a circular banked turn. This shows the velocity vector as well

lunch2 on turntable.py
Shows the velocity and acceleration of and the gravitational and normal force and static frictional force on a box sitting on a horizontal rotating turntable.

Horizontal SHM.py
Improved version of **spring_constant4** showing position vs time graph and clearer labels

dm_segment.py shows a small segment of arc in a rotating ring (part of moment of inertia of a ring calculation)

disk_sphere_2.py
Illustrates the disk being used for moment of inertia calculations for a uniform sphere

shell_sphere_extrude.py
Illustrates using the shell method to calculate the moment of inertia for a uniform sphere

slice of spherical shell.py
Illustrates the shape of the differential ring used to calculate the moment of inertial of a thin, hollow, spherical shell

rotating disk.py
Illustrates how points at different radii on a rotating disk have a different tangential speed

ring_around a diameter try5.py
Illustrates in 3D how the differential ring we use to calculate the moment of inertia of a shell fits into the overall shape of the shell

shell_sphere_extrude.py
Shows the *rotating* differential shell used to calculate the moment of inertia of a sphere. The challenge here is that the way I drew the shell (using an extrusion) doesn't have a built in way to rotate it. (There is a "rotate" command built in for other simple shapes.)

ring_around a diameter try2 no dm.py
Attempts to show a ring rotating around a diameter (again involves trying to get an extrusion to rotate)

parallel and perpendicular ring.py
Draws three rings that are mutually perpendicular, to illustrate an example problem that I want students to solve.

ring_around a diameter try2.py
Shows a ring rotating around a diameter (again involves trying to get an extrusion to rotate), and a section of the ring with mass *dm* that we would use for an integration.

disk around diameter.py
Shows the differential mass element (a ring) that we would use to integrate over a disk spinning around a diameter to calculate its moment of inertia, and the series of mass elements *dm* that we will be integrating over.

I sphere by disk method.py
Another attempt to show the differential mass element (a disk) that we could use to calculate the moment of inertia of a sphere, and then animates the series of disks that we would be integrating over.

I pendulum.py
This is for a physical pendulum, for a disk at the end of a rod. It shows how the rotation axis of the disk has been displaced to the end of the rod

pt mass in circle.py
Shows a mass *m* moving at speed v in a circle of radius R

I of box demo.py
Shows two identical boards, and draws two different rotation axes—one through the middle along the length of the board, and another through the middle but along the short width of the board.

board and pulley.py
Shows these two boards, each with a pulley attached and a string wrapped around the pulley attached to identical hanging masses. The board with the axis along its length has a smaller moment of inertia than the other board, so that board rotates faster and the mass descends faster.

torque off center impulse demo2.py
Shows three identical rods in space. One gets an impulse at its center, the second slightly off center, and the third more off center.

pulley.py
Attempt to draw a pulley using an extrusion

pulley for torque intro.py
Identical disks, mounted on a horizontal axis. Each has a pulley mounted on the disk, but for one disk the pulley is twice the diameter of the other. Same pulling force on each disk. Illustrates how torque diameter affect the angular acceleration of the disk, all else being equal.

cow2.py
Draws a cow(!) from various other standard shapes. Shows the cow rotating as it moves horizontally at constant speed through space. First time I used a "frame", which takes a number of individual elements and combines them into a single element whose position and rotation can be controlled as a single object rather than having to move each object individually.

cow3.py
Shows a cow in space with a single force acting on its center of mass; with two equal and opposite forces acting on its center of mass; and finally two equal and opposite forces offset from the center of mass

try arc extrusion.py
Draws a helix using some extrusion commands

rotation vectors rev3.py
Illustrates how rotations of increasingly larger angles can be represented by increasing longer axial vectors

demo omega from rotational disp vector.py
Shows how the angular displacement vector gets longer at a constant rate when a disk spins at a constant angular speed

pulley for torque intro change F.py
Identical disks, mounted on a horizontal axis. Each has an identical pulley mounted on the disk, with a string and hanging weight wrapped around the pulley, but one weight is twice as large as the other. Illustrates how increasing the pulling force increases the angular acceleration of the disk, all other factors remaining equal.

demo alpha from rotational omega vector.py
Shows how the angular velocity vector changes as a disk speeds up or slows down its rotation speed or changes direction.

omega hanging mass on pulley.py
Shows a disk mounted on a horizontal axis. It has a pulley mounted on the disk, with a string and hanging weight wrapped around the pulley. First shot at showing how the angular velocity vector of a disk changes as the hanging weight descends, then starts to rewind and the hanging weight rises after the weight gets to the end of the string.

omega hanging mass on pulley rev 3.py
Same as above but shows the angular acceleration vector and rotates the perspective of the disk as it turns so you can see the vector on the back side of the disk when the disk changes direction. Also graphs the angular velocity as a function of time, in real time with the spinning of the disk.

v as omega x r.py illustrates the relationship between the velocity, radius, and angular velocity vectors as a cross product for a mass going in a circle at constant speed.

pulley for tau and alpha direction question.py Shows a disk mounted on a horizontal axis. It has a pulley mounted on the disk, with a string and hanging weight wrapped around the pulley. Asks students to draw the direction for the radius, force torque and angular acceleration vectors for the situation where the mass is

a) descending and disk is spinning clockwise
b) ascending and disk is spinning clockwise
c) descending and disk is spinning counterclockwise
d) ascending and disk is spinning counterclockwise

rotation bar.py illustrates how points at different points on a rotating bar have a different tangential speed

Simple pendulum for SHM restoring forces init equil.py starts with pendulum hanging straight down, then the forces and components of force acting on it when it is pulled to one side, the same when it is pulled to the other, and then its oscillating motion and the net force acting on after it is released.

expt SHM projection on a circle with drag.py Has a point on a circle, initially on the x axis. You drag it to some other point on the circle and then let go. The mass continues in uniform circular motion from there, and the x-coordinate of its motion is graphed at the same time, so you can see how changing the starting phase changes the graph of the displacement with time.

model_rocket_rev2 (animates the launch of a "rocket" that is under power for 4 seconds, runs out of fuel and continues to rise and slow down, then falls back to the ground. Generates the position, velocity, and acceleration vs. time graphs in "real time" as the rocket rises and falls) This is part of an example problem that appears in one of the videos lectures on constant acceleration.

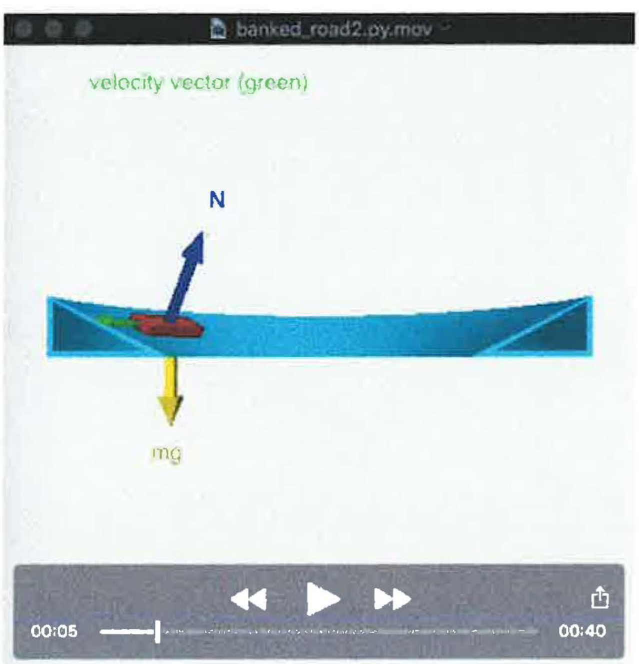
banked_road2.py

banked_road2.py A problem we solve involves a car making a turn on a banked road without friction. This illustrates the gravitational and normal forces acting on the car as it moves along a circular banked turn. This shows the velocity vector as well

banked_road2.py - /Users/pwolf/Documents/Sabbatical Related Stuff/Learning VPython/banked_road2.py (2.7.9)

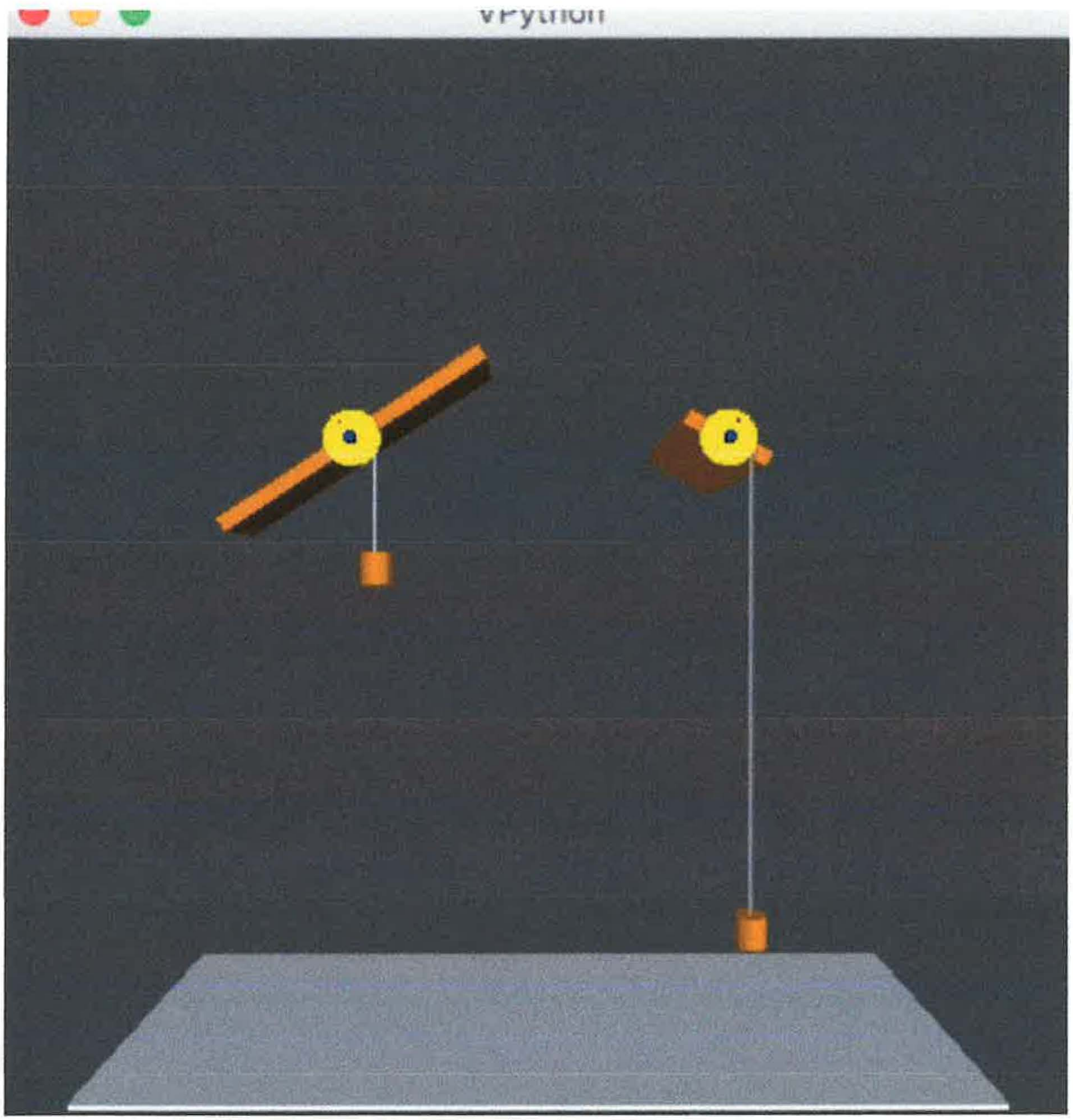
```
from visual import *
from time import *
from math import pi, sin, cos

scene = display (background = (0.9,0.9,0.9))
bank = Polygon( [(0,0), (40,0), (40,20)] )
semicircle = paths.arc(radius=50, angle2=pi)

in_bank = Polygon([(8,2),(38,2),(38,18)])

road = extrusion(pos=semicircle, shape=bank-in_bank,
         color=(0,0.8,1))
car = box (pos = (68,12,0), axis = (-2,4,0), color = color.red, length = 6, width = 30, height = 10)
car_ax = arrow(pos = car.pos, axis = (-20, 40,0), color = color.blue, shaftwidth = 4)
mg_arrow = arrow (pos = car.pos +(0,0,-5), color = color.yellow, axis = (0,-40,0), shaftwidth = 4)
N_lbl = label (pos = car.pos + (-25, 45,0), color = color.blue, text = 'N', box = False, line = False)
mg_lbl = label (pos = car.pos - (0,50,0), text = 'mg', color = (0.6, 0.6,0), box = False, line = False)
scene.waitfor('click')
vel_pointer = arrow (color = (0, 0.9, 0), axis = (0, 0, -1), length = 20, pos = (68*cos(0.25), 12, -68*sin(0.25)))

'''
car.pos = (68*cos(60*pi/180), 12,-68*sin(60*pi/180))
car.axis = (-2*cos(60*pi/180), 4, 2*sin(60*pi/180))
car.rotate (angle = 60*pi/180)
car_ax.pos = car.pos
car_ax.axis = car.axis
car_ax.length = 20
'''

vel_label = label (pos = (-40,90,0), color = (0, 0.8, 0), box = False, line = False, text = 'velocity vector (green)')
for i in range(1440):
    rate(40)
    car.pos = (68*cos(i*pi/180.0), 12,-68*sin(i*pi/180.0))
    car.axis = (-2*cos(i*pi/180.0), 4, 2*sin(i*pi/180.0))
    car.rotate (angle = pi/201, origin = car.pos, axis = car.axis)
    car_ax.pos = car.pos
    car_ax.axis = car.axis
    car_ax.length = 40
    mg_arrow.pos = car.pos +(0,0,-5)
    mg_lbl.pos = car.pos + (0,-50,0)
    N_lbl.pos = car.pos + (-25*cos(i*pi/180.0),45,25*sin(i*pi/180.0))
    vel_pointer.pos = (68*cos(0.25+i*pi/180.0), 12,-68*sin(0.25+i*pi/180.0))
    vel_pointer.axis = (-20*sin (i*pi/180), 0, -20*cos(i*pi/180))


scene.waitfor('click')
exit()
```

Ln: 23 Col: 11



board and pulley.py

board and pulley.py Shows these two boards, each with a pulley attached and a string wrapped around the pulley attached to identical hanging masses. The board with the axis along its length has a smaller moment of inertia than the other board, so that board rotates faster and the mass descends faster.



Screenshot of one frame from board and pulley.py board and pulley.py - /Users/pwolf/Documents/Sabbatical Related Stuff/board and pulley.py (2.7.9)

```
from visual import *
from math import *
from time import *
scene.background = (0.3,0.3,0.3)
floor = box(pos=(0,-280,0), size = (500,3,200), material = materials.marble)

'''
quantities for problem:
pulley diameter = 0.05 m
board dimensions = 1m long, 30 cm wide, 5 cm high
mass of board = 5 kg
g = 9.8 N/kg
'''
m_hang = 0.2
g = 9.9
R = 0.02
m1 = 5
W = 0.3
L=1.0
omega = 0
theta = 0
omega2 = 0
theta2 = 0
t = 0
dt = 0.05


#draw Pulley
rect = Polygon([(0,-5),(0,5),(40,5),(40,4),(37,3),(37,-3),(40,-4),(40,-5)])
#p = paths.circle(pos = (0,0,0), radius = 3)
#pulley = extrusion(pos =p, shape = rect, color = color.yellow)
#draw the first board, pulley, mass, etc.
board1 = box(pos=(120,90,-100), size = (60,10,200), color = color.orange)
spindle1 = cylinder(pos =(120,90,0), axis = (0,0,7), radius = 4, color = color.blue)
#draw vertical pulley
pul_2 = Polygon([(0,-3),(0,3),(15,3),(15,2),(13,1),(13,-1),(15,-2),(15,-3)])
r = paths.circle(pos = (120,90,0), radius = 3, up=(0,0,1))
pulley = extrusion(pos =r, shape = pul_2, color = color.yellow)
pulley_spot = cylinder (pos = (120,102,0), axis = (0,0,7), color = color.red)
#draw vertical string
v_string=cylinder(pos = (135,90,0), radius = 1, material = materials.marble, axis = (0,-50,0))
hang_m = cylinder(pos = v_string.pos+(0,-50,0), radius = 10, color = color.orange, axis = (0,-20,0))


#draw the second board, pulley, mass, etc.
board2 = box(pos=(-120,90,-30), size = (200,10,60), color = color.orange)
spindle2 = cylinder(pos=(-120,90,0), axis = (0,0,7), radius = 4, color = color.blue)
#draw vertical pulley
pul_2 = Polygon([(0,-3),(0,3),(15,3),(15,2),(13,1),(13,-1),(15,-2),(15,-3)])
r2 = paths.circle(pos = (-120,90,0), radius = 3, up=(0,0,1))
pulley2 = extrusion(pos =r2, shape = pul_2, color = color.yellow)
pulley_spot2 = cylinder (pos = (-120,102,0), axis = (0,0,7), color = color.red)
#draw vertical string
v_string2=cylinder(pos = (-105,90,0), radius = 1, material = materials.marble, axis = (0,-50,0))
hang_m2 = cylinder(pos = v_string2.pos+(0,-50,0), radius = 10, color = color.orange, axis = (0,-20,0))
```

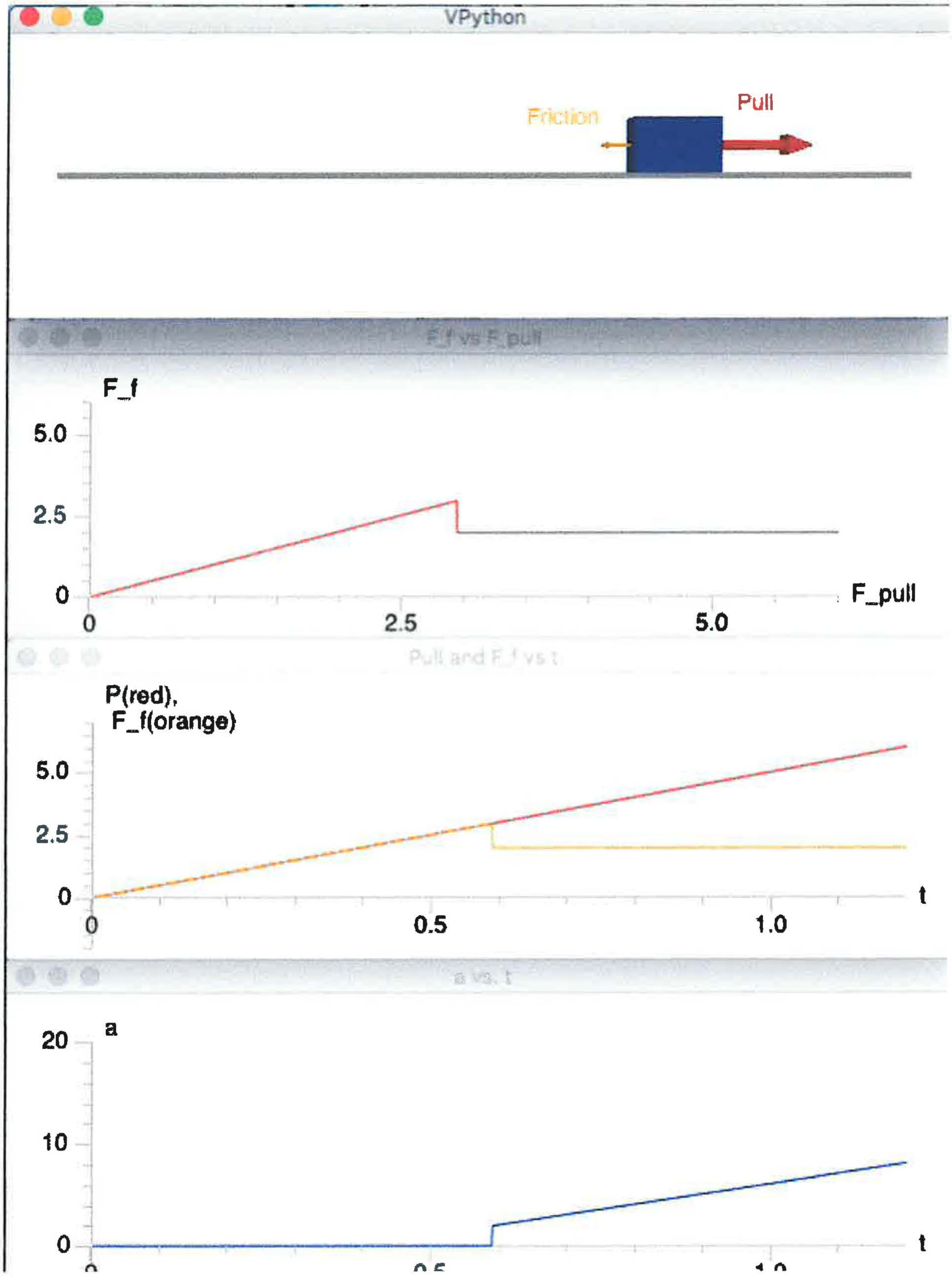
(page 1 of board and pulley.py code. Continued on the next page.)

```
#draw the second board, pulley, mass, etc.
board2 = box(pos=(-120,90,-30), size = (200,10,60), color = color.orange)
spindle2 = cylinder(pos=(-120,90,0), axis = (0,0,7), radius = 4, color = color.blue)
#draw vertical pulley
pul_2 = Polygon([(0,-3),(0,3),(15,3),(15,2),(13,1),(13,-1),(15,-2),(15,-3)])
r2 = paths.circle(pos = (-120,90,0), radius = 3, up=(0,0,1))
pulley2 = extrusion(pos =r2, shape = pul_2, color = color.yellow)
pulley_spot2 = cylinder (pos = (-120,102,0), axis = (0,0,7), color = color.red)
#draw vertical string
v_string2=cylinder(pos = (-105,90,0), radius = 1, material = materials.marble, axis = (0,-50,0))
hang_m2 = cylinder(pos = v_string2.pos+(0,-50,0), radius = 10, color = color.orange, axis = (0,-20,0))


while hang_m2.y>-250:
    rate(20)
    #board 1
    if hang_m.y>-250:
        torque = m_hang*g*R
        I_1 = 1/12.0*m1*W**2
        alpha1 = torque/I_1
        d_omega = alpha1*dt
        omega +=d_omega
        d_theta = (omega -d_omega/2.0)*dt
        theta += d_theta
        board1.rotate(angle = -d_theta, axis = (0,0,1), origin = (120, 90,0))
        pulley_spot.rotate (angle = -d_theta, axis = (0,0,1), up = (0,0,1), origin = (120,90,0))
        dy=R*d_theta
        hang_m.y -=200*dy
        v_string.axis.y -=200*dy

    #board
    torque = m_hang*g*R
    I_1 = 1/12.0*m1*L**2
    alpha2 = torque/I_1
    d_omega2 = alpha2*dt
    omega2 +=d_omega2
    d_theta2 = (omega2 -d_omega2/2.0)*dt
    theta2 += d_theta2
    #print 'alpha=', alpha1,'omega=', omega, 'theta =', theta
    board2.rotate(angle = -d_theta2, axis = (0,0,1), origin = (-120, 90,0))
    pulley_spot2.rotate (angle = -d_theta2, axis = (0,0,1), up = (0,0,1), origin = (-120,90,0))
    dy2=R*d_theta2
    hang_m2.y -=200*dy2
    v_string2.axis.y -=200*dy2
    t +=dt


scene.waitfor('click')
exit()
```

(page 2 of board and pulley.py code.)

Friction demo.py

Friction_demo.py shows the motion of a block on a sticky horizontal surface as the pulling force slowly increases. Graphs the friction force, pulling force, and acceleration of the block as a function of time



Friction demo.py - /Users/pwolf/Documents/Sabbatical Related Stuff/Learning VPython/Friction demo.py (2.7.9)

```python
from visual import *
from time import *
from visual.graph import *


scene = display (background = (1,1,1), width = 600, height = 200)

#define graph characteristics:
# F_f vs F_pull
gd1 = gdisplay(x=0, y=201, width=600, height=200,
      title='F_f vs F_pull', xtitle='F_pull', ytitle='F_f',
      foreground=color.black, background=color.white,
      xmax=6, xmin=0, ymax=6, ymin=0)
FvsP = gcurve(color=color.red, gdisplay = gd1) # a connected curve object


#a vs. t graph
gd3 = gdisplay(x=0, y=601, width=600, height=200,
      title='a vs. t', xtitle='t', ytitle='a',
      foreground=color.black, background=color.white,
      xmax=1.2, xmin=0, ymax=20, ymin=-2)

avst = gcurve(color = color.blue, gdisplay = gd3)# a connected curve object

#F, F_f vs. t graph

gd2 = gdisplay(x=0, y=401, width=600, height=200,
      title='Pull and F_f vs t', xtitle='t', ytitle='P(red), \n F_f(orange)',
      foreground=color.black, background=color.white,
      xmax=1.2, xmin=0, ymax=7, ymin=-2)
Pvst = gcurve(color = color.red, gdisplay = gd2)
Ffvst = gcurve(color = color.orange, gddisplay = gd2)

#draw track
#right/left of track at +/-1500; top/bottom at +/-10; front back at +/-100
track = box(pos = (0,0,0), color = (0.6,0.6,0.6), length = 3000, height = 20, width = 200)


#draw cart--box centered at (-1200,75,0)
#cart front/back at -1350/-1150; top/bottom at 125/25; front/back at 90/-90
# wheels centers 40 from each end and level with bottom of cart.
# Inner face of each wheel is 80 from center of track, and is 6 wide.
cart = box(pos = (-1200,110,0), color = color.blue, length = 300, height = 200, width = 180)

F_pull = 0
F_pull_arrow = arrow (pos = cart.pos + (150,0,0), axis = (F_pull,0,0), color = color.red)
Pull_label = label(pos = F_pull_arrow.pos + (350,150,0),box = False, height = 12, color = color.red, text = "Pull")

F_f = 0
F_f_arrow = arrow (pos = cart.pos - (150,0,0), axis = (-F_f,0,0), color = color.orange)
F_f_label = label(pos = F_f_arrow.pos + (-350,100,0), box = False, height = 12, color = color.orange, text = "Friction")

#define the wall at the end of the track and the spring on it
#initial position of the spring end facing the cart is (1100,155,0)
'''

anchor = box(pos = (1505,120,0), length = 10, height = 250, width = 200, color = (0.6,0.6,0.6))
spring = helix(pos = (1500, 155,0), length = 400, radius = 25,
         color = (0.6,0.6,0.6), coils = 8, thickness = 8, axis = (-1,0,0))
'''
```

Ln: 118 Col: 0

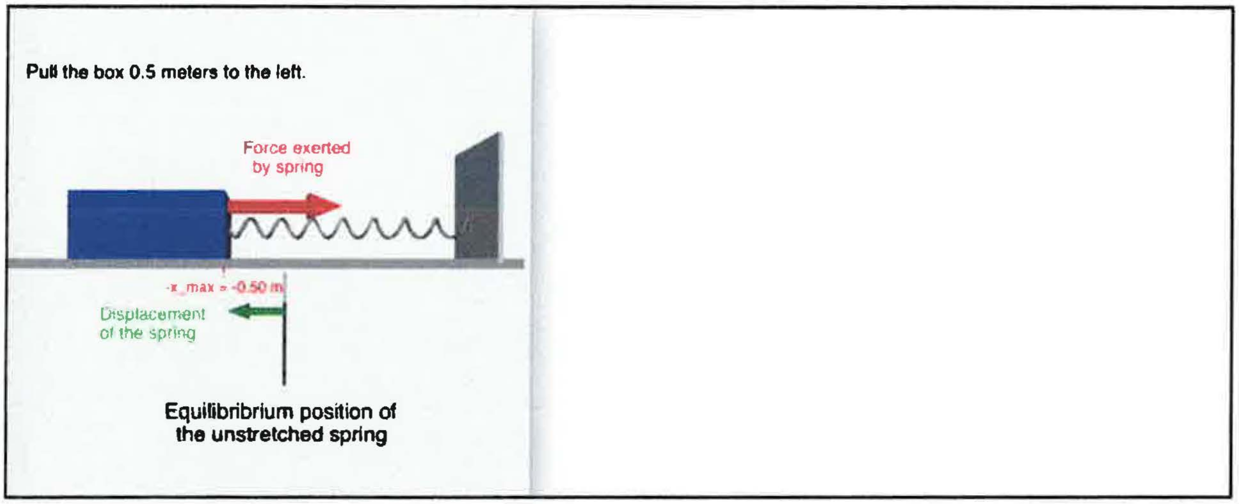
Friction demo.py - /Users/pwolf/Documents/Sabbatical Related Stuff/Learning VPython/Friction demo.py (2.7.9)

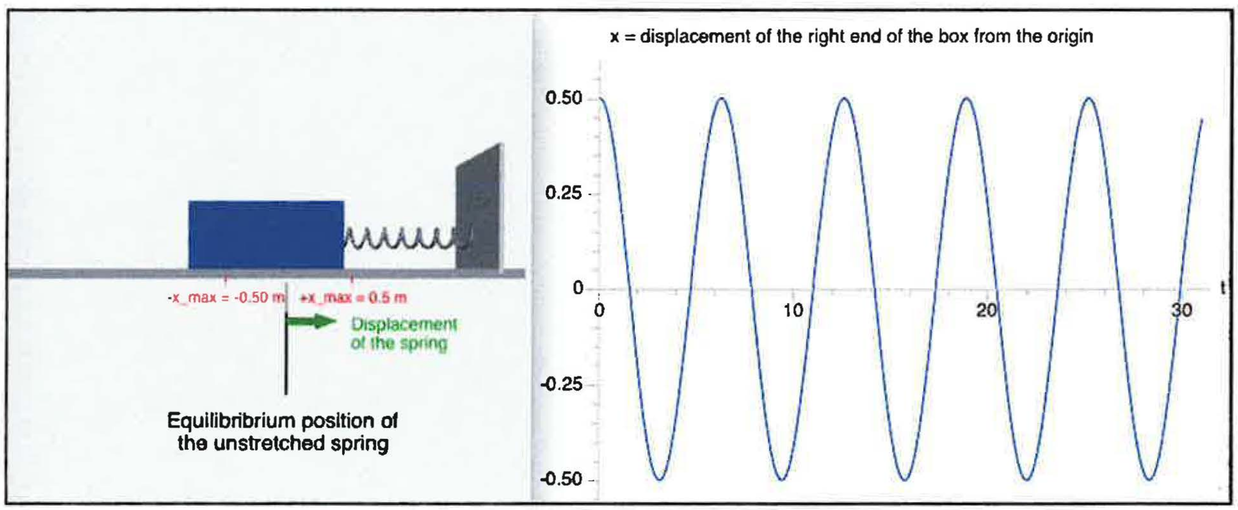
```
#x0 = -975/scalefactor  #-980 is the initial position of the front of the stopper

a = 0
v = 0
dx = 0
dv = 0
dt = 0.001
x = cart.x
t = 0
u_s = 0.60
u_k = 0.4
m = 0.5
g = 9.8
dF_pull = 0.005
while F_pull < u_s*m*g:
    rate(50)
    F_pull += dF_pull
    F_f += dF_pull
    F_pull_arrow.axis = (F_pull*60,0,0)
    Pull_label.x = F_pull_arrow.x + 150
    F_f_arrow.axis = (-F_pull*60,0,0)
    F_f_label.x = F_f_arrow.x -250
    FvsP.plot(pos = (F_pull,F_pull))
    t +=dt
    v = 0
    avst.plot(pos = (t,a))
    Pvst.plot(pos= (t, F_pull))
    Ffvst.plot(pos= (t, F_f))

while F_pull < 6:
    rate(50)
    a = (F_pull-u_k*m*g)/m
    F_f = u_k*m*g
    dv = a*dt
    v += dv
    dx = (v+dv/2.0)*dt
    x +=dx
    t +=dt
    cart.x +=dx*2500
    F_pull_arrow.x = cart.x + 150
    Pull_label.x = F_pull_arrow.x + 150
    F_f_arrow.x = cart.x - 150
    F_f_arrow.axis = (-F_f*60,0,0)
    F_f_label.x = F_f_arrow.x -250
    F_pull_arrow.axis = (F_pull*60,0,0)
    FvsP.plot(pos = (F_pull,u_k*m*g))
    F_pull +=dF_pull
    avst.plot(pos = (t,a))
    Pvst.plot(pos= (t, F_pull))
    Ffvst.plot(pos= (t, F_f))


scene.waitfor('click')
exit()
```

Ln: 118 Col: 0

Horizontal SHM.py

Horizontal SHM.py
Improved version of **spring_constant4** showing position vs. time graph and clearer labels (Horizontal mass spring system showing the system at equilibrium, then the mass is pulled to one side and released and starts oscillating. There are arrows that change length and location showing the force and displacement with time and labels that move with them.)





(Screenshots from Horizontal SHM.py)

Horizontal SHM.py - /Users/pwolf/Documents/Sabbatical Related Stuff/Learning VPython/Horizontal SHM.py (2.7.9)

```
#from vpython import *
from visual import *
from visual.graph import *
from time import *
from math import sin, cos, pi, sqrt
scene = display(background = (0.9,0.9,0.9), foreground = color.black)

#define graph characteristics:
# x vs t  graph
gd1 = gdisplay(x=455, y=0, width=600, height=450,
      title='xvs. t', xtitle='t', ytitle='x = displacement of the right end of the box from the origin',
      foreground=color.black, background=color.white,
      xmax=10*pi, xmin=0, ymax=0.6, ymin=-0.6)
xvst = gcurve(color=color.blue, gdisplay = gd1) # a connected curve object


#draw track
#right/left of track at +/-1500; top/bottom at +/-10; front back at +/-100
track = box(pos = (0,0,0), color = (0.6,0.6,0.6), length = 1000, height = 20, width = 200)

#makes a frictionless cart
cart = box(pos = (-100,75,0), color = color.blue, length = 300, height = 130, width = 180)


# Draw spring and anchor
spring = helix(pos = cart.pos +(550,0,0), length = 400, radius = 20,
         color = (0.6,0.6,0.6), coils = 8, thickness = 8, axis = (-1,0,0))
anchor = box(pos = spring.pos + (5,50,0), length = 10, height = 250, width = 200, color = (0.8,0.8,0.8))

#Unstretched length label
equil_label = label (pos = cart.pos+(150, -410,0), box = False, line = False, height = 15, text = 'Equilibribrium position of \n  the unstretched spring')
eq_pointer = arrow (pos = equil_label.pos+(0,80,0), axis = (0,240,0), shaftwidth = 5)
scene.waitfor('click')

F_spring = 0
del_x = 0
k = 2
m = 2
omega = sqrt(k/m)


#Arrow for force of spring on cart
what_haps_label = label (pos = (-200,400,0), text = 'Pull the box 0.5 meters to the left.', box = False)
Force_arrow = arrow( pos = cart.pos +(150, +45,0), axis = (F_spring,0,0), box = False, color = color.red)
Force_text = label(pos = Force_arrow.pos + (150,100,0), box = False, height = 12,color = color.red, text = 'Force exerted \n  by spring', opacity = 0)
del_x_arrow = arrow(pos = (50, -100,0), axis = (-del_x,0,0), box = False, color = (0,0.6,0), shaftwidth = 15)
del_x_text = label(pos = Force_arrow.pos + (-150,-250,0), box = False, height = 12, color = (0,0.6,0), text = 'Displacement \nof the spring')


t = 0
while (t/60.0)<(pi/2.0):
    rate (40)
    del_x = -120.0 * math.sin(t/60.0) #Sinusoidal, starting to the left
    cart.x = -100+del_x     # x = -100 is the initial position of the center of the cart
    Force_arrow.x = 50 + del_x  #cart is 300 long, center at -100, so left edge is at 50
    del_x_arrow.axis = (del_x,0,0)  # determines direction and length of displacement arrow
    F_spring = k*del_x
    Force_arrow.axis = (-F_spring,0,0)
    spring.length =400-del_x
    if del_x < 0:
        Force_text.x = cart.pos.x +300
        del_x_text.x = cart.pos.x
    else:
        Force_arrow.x = cart.x + 150 +F_spring #cart.x + 150 is right edge of cart. Want tail to be F_spring further to right
        Force_arrow.axis = (-F_spring,0,0)  #arrow points to left, has length F_spring. Tip is on right edge of cart
        Force_text.x = cart.pos.x +50
        del_x_text.x = cart.pos.x +300
    t +=1
```

Ln: 14 Col: 67

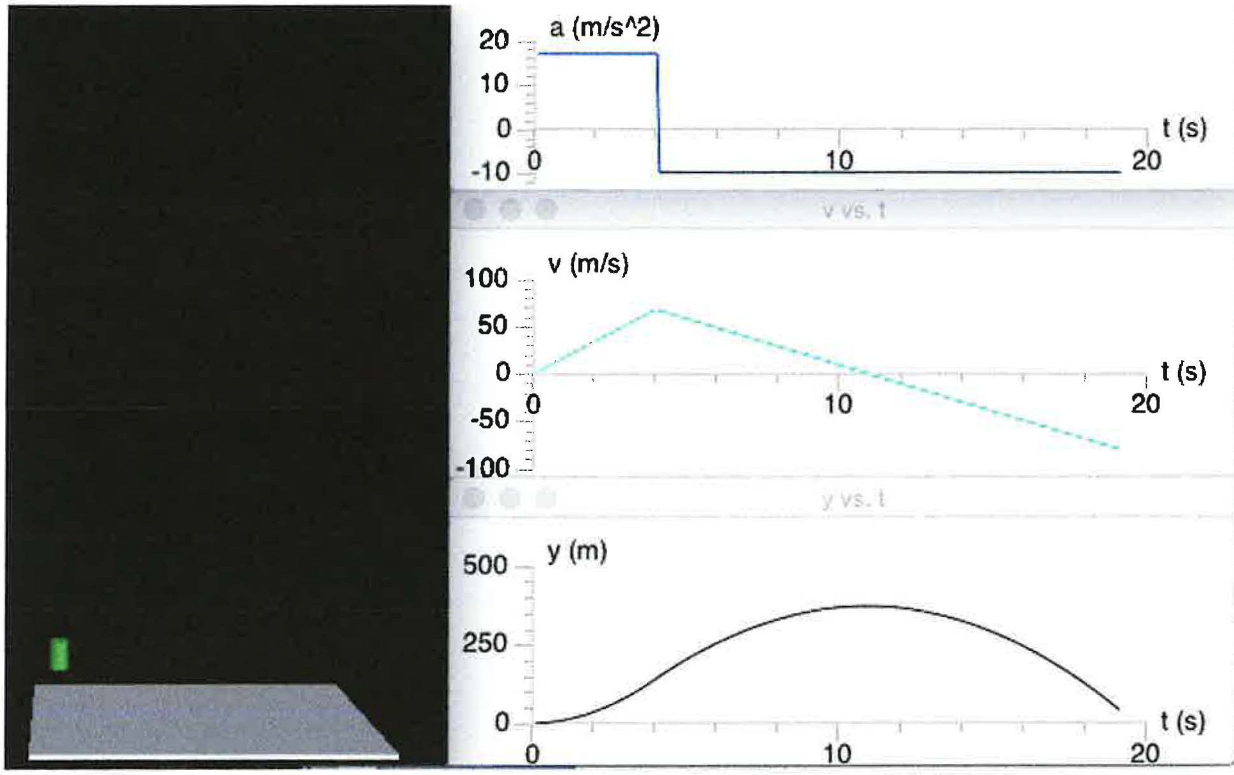
(page 1 of code for Horizontal SHM.py)

```
neg_x_max_label = label(pos = cart.pos+(cart.length/2, -120,100), box = False, height = 10, text = '-x_max = -0.50 m', color = color.red)
neg_x_max_marker = arrow(pos = neg_x_max_label.pos + (0,20,0), axis = (0,20,0), color = color.red)
scene.waitfor('click')
what_haps_label.visible = False
what_haps_label.text = "If you let go of the mass it will have \n the same max displacement \n on the other side of the origin"
what_haps_label.visible = True

while (t/60.0)<(3*pi/2.0):
    rate (40)
    del_x = -120.0 * math.sin(t/60.0) #Sinusoidal, starting to the left
    cart.x = -100+del_x    # x = -100 is the initial position of the center of the cart
    Force_arrow.x = 50 + del_x  #cart is 300 long, center at -100, so left edge is at 50
    del_x_arrow.axis = (del_x,0,0)  # determines direction and length of displacement arrow
    F_spring = k*del_x
    Force_arrow.axis = (-F_spring,0,0)
    spring.length =400-del_x
    if del_x < 0:
        Force_text.x = cart.pos.x +300
        del_x_text.x = cart.pos.x
    else:
        Force_arrow.x = cart.x + 150 +F_spring #cart.x + 150 is right edge of cart. Want tail to be F_spring further to right
        Force_arrow.axis = (-F_spring,0,0)  #arrow points to left, has length F_spring. Tip is on right edge of cart
        Force_text.x = cart.pos.x +300
        del_x_text.x = cart.pos.x +300
    t +=1

x_max_label = label(pos = cart.pos+(cart.length/2, -120,100), box = False, height = 10, text = '+x_max = 0.5 m',color = color.red)
x_max_marker = arrow(pos = x_max_label.pos + (0,20,0), axis = (0,20,0), color = color.red)
what_haps_label.visible = False
Title = label (pos = (-200,400,0), text = "Catch the mass at this point.\n Then let go of the mass.\nCall this t = 0.")

scene.waitfor('click')
Force_arrow.visible = False
Force_text.visible = False
t = 0
Title.visible = False
'''#define graph characteristics:
# x vs t. graph
gd1 = gdisplay(x=480, y=0, width=600, height=400,
      title='xvs. t', xtitle='t', ytitle='x',
      foreground=color.black, background=color.white,
      xmax=1000, xmin=0, ymax=0.6, ymin=-0.6)
xvst = gcurve(color=color.cyan, gdisplay = gd1) # a connected curve object
'''
while t<10*pi*omega:
    rate (60)
    del_x = 120.0 * math.cos(omega*t)
    cart.pos.x = del_x-100 #cosinusoidal, starting to the left

#   Force_arrow.x = 50 + del_x  #cart is 300 long, center at -100, so left edge is at 50
    del_x_arrow.axis = (del_x,0,0)  # determines direction and length of displacement arrow
    F_spring = k*del_x
#   Force_arrow.axis = (-F_spring,0,0)
    spring.length =400-del_x
    if del_x < 0:
#       Force_text.x = cart.pos.x +300
        del_x_text.x = cart.pos.x
    else:
#       Force_arrow.x = cart.x + 150 +F_spring #cart.x + 150 is right edge of cart. Want tail to be F_spring further to right
#       Force_arrow.axis = (-F_spring,0,0)  #arrow points to left, has length F_spring. Tip is on right edge of cart
#       Force_text.x = cart.pos.x +50
        del_x_text.x = cart.pos.x +300
    xvst.plot(pos = (t,del_x/240.0))
    t+=1/60.0


scene.waitfor('click')
exit()
```

(page 2 of code for Horizontal SHM.py)

model_rocket_rev2.py

model_rocket_rev2 (animates the launch of a "rocket" that is under power for 4 seconds, runs out of fuel and continues to rise and slow down, then falls back to the ground. Generates the position, velocity, and acceleration vs. time graphs in "real time" as the rocket rises and falls)
This is part of an example problem that appears in one of the videos lectures on constant acceleration.



(Screenshot from model_rocket_rev2.py)

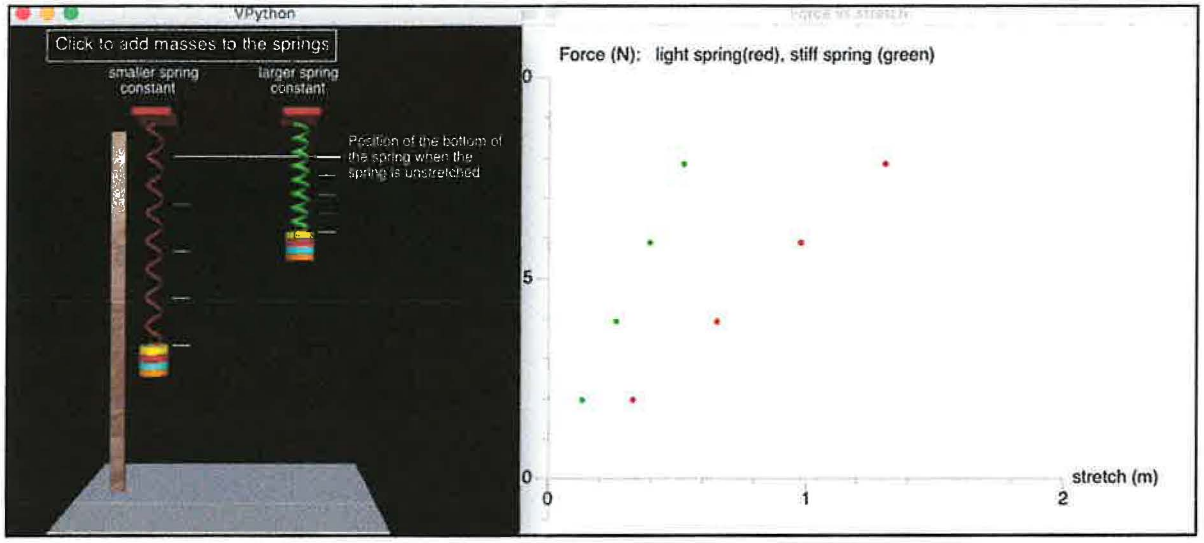
VPython Code for model_rocket_rev2.py

Model_rocket_rev2.py - /Users/pwolf/Documents/Sabbatical Related Stuff/Lea...

```
from visual import *
from math import *
from visual.graph import * # import graphing features
from numpy import arange, cos, exp
import time



# vvs t. graph
gd1 = gdisplay(x=450, y=130, width=450, height=160,
      title='v vs. t', xtitle='t (s)', ytitle='v (m/s)',
      foreground=color.black, background=color.white,
      xmax=20, xmin=0, ymax=100, ymin=-100)
vvst = gcurve(color=color.cyan, gdisplay = gd1) # a connected curve object

gd2 = gdisplay(x=450, y=290, width=450, height=160,
      title='y vs. t', xtitle='t (s)', ytitle='y (m)',
      foreground=color.black, background=color.white,
      xmax=20, xmin=0, ymax=500, ymin=0)
yvst = gcurve(color = color.black, gdisplay = gd2)


gd3 = gdisplay(x=450, y=0, width=450, height=130,
      title='a vs. t', xtitle='t (s)', ytitle='a (m/s^2)',
      foreground=color.black, background=color.white,
      xmax=20, xmin=0, ymax=20, ymin=-12)
avst = gcurve(color = color.blue, gdisplay = gd3)



scene.background= (0.1,0.1,0.1)

v0 = 0 #m/s
v_y = v0
t = 0
dt = 0.1
y_0 = -120
y = y_0
ground = box(pos = (50,-122,0), length = 120, width = 50, height = 2)
rocket = cylinder(pos = vector(0,-120,0), radius = 3, axis = (0,10,0), color = color.gree
scale = 0.4
rocket.y=y_0

while t < 4:
    a = 17
    rate(10)
    t += dt
    v_y += a*dt
    dy = v_y * dt
    y += dy
    rocket.y += dy*scale
    avst.plot(pos = (t, a))
    vvst.plot(pos = (t,v_y))
    yvst.plot(pos = (t, y+120))

while y > -120:
    a = -9.8
    rate(10)
    t += dt
    v_y += a*dt
    dy = v_y * dt
    rocket.y += dy*scale
    y += dy
    avst.plot(pos = (t, a))
    vvst.plot(pos = (t,v_y))
    yvst.plot(pos = (t, y+120))


rocket.y = -120
scene.waitfor('click')
exit()
```

Ln: 57 | Col: 14

Spring_Constant3_IDLE.py (Illustrates a lab with two different hanging springs, and plots the stretch vs. hanging weight as progressively more weight is hung on each spring.)



Spring_Constant3_IDLE.py - /Users/pwolf/Documents/Sabbatical Related Stuff/Learning VPython/Spring_Consta...

```
#from vpython import *
from visual import *
from visual.graph import *
from time import *
#import graphics
#import random

scene = display(background = (0.1,0.1,0.1))

#define graph characteristics:
# height vs t. graph
gd1 = gdisplay(x=430, y=0, width=600, height=450,
      title='Force vs.stretch', xtitle='stretch (m)', ytitle='Force (N):   light spring(red), stiff spring (green)',
      foreground = color.black, background=color.white,
      xmax=2, xmin=0, ymax=10, ymin=-1)
Fvsx = gdots(color=color.red, gdisplay = gd1) # a connected curve object

#defines floor that the motion sensor is sitting on
floor = box(pos = vector(-50,-292,0), length = 400, width = 200, height = 4,
        color = color.white, axis=vector(1,0,0))

#makes a ruler
ruler = box(pos = (-200,-40,-0), axis = vector(0,0,1), length = 4, width = 20, height = 500, material = materials.wood)


#make the first anchor, spring, and indicator of bottom of spring
anchor = box(pos = vector(-150,230,0), length = 50, width = 50, height = 10,
        color = color.red, axis=vector(1,0,0))
spring = helix(pos = anchor.pos - (0,5,0), length = 50, radius = 10,
        color = color.red, coils = 8, thickness = 4, axis = vector(0,-1,0))
unstretched_line = cylinder(pos=spring.pos - (10,0,175), axis = vector(300,0,0), radius = 1, color = color.white)
spring_label = label(pos = anchor.pos + (0,50,0), box = False, height = 10, text = "smaller spring \n   constant")


#make the second anchor, spring, and indicator of bottom of spring
newanchor = box(pos = vector(50,230,0), length = 50, width = 50, height = 10,
        color = color.red, axis=vector(1,0,0))
newspring = helix(pos = newanchor.pos - (0,5,0), length = 50, radius = 10,
        color = color.green, coils = 8, thickness = 7, axis = vector(0,-1,0))
newunstretched_line = cylinder(pos=newspring.pos - (-45,0,175), axis = vector(50,0,0), radius = 1, color = color.white)
newspring_label = label(pos = newanchor.pos + (0,50,0), box = False, height = 10, text = "larger spring \n   constant")


directions_label = label(pos = anchor.pos + (50,100,0), line = False, text = "Click to add masses to the springs")


unstretched_label = label(pos = newunstretched_line.pos, xoffset = 20, box = False, height = 10,
                          text = 'Position of the bottom of \nthe spring when the\nspring is unstretched')


#makes a new anchor to attach the new spring to

#defines new unstretched spring, pointing down from the anchor

#marker for unstretched length of the spring

scene.waitfor('click')
```

```python
#add mass to spring
massHeight = 10

mass = cylinder (pos = spring.pos - (0,50,0), axis = vector(0,-massHeight,0), radius = 20, color = color.yellow)
#directions_label.pos += (0,-50,0)

#define parameters for the problem
m = 0
delm = 0.20  #kg--the mass
k = 6.0 #N/m--the spring constant
k2 = 15
g = 9.8  #N/kg
scalefactor = 200 #scales the displacements so that they show up on the program display

#add mass to first spring
for i in range(4):
    m +=delm
    # calculation for first sprint
    stretch = scalefactor* (m*g/k)  #stretch of the spring
    delstretch = scalefactor* delm*g/k
    spring.length += delstretch  #calculate the new length of the spring
    mass.pos +=(0,-delstretch,0)
    #newmass.length += 5   from when the cylinder just stayed the same color but got longer
    if i==1:  # add the second mass to the bottom of the first one
        addedmass = cylinder(pos = mass.pos - (0,massHeight,0), axis = vector(0,-massHeight,0), radius = 20, color = color.red)
    if i==2:  #add third mass to the bottom of the second one
        addedmass.pos +=(0,-delstretch,0)  #move second mass to new, lower position
        addedmass2 = cylinder(pos = mass.pos - (0,2*massHeight,0), axis = vector(0,-massHeight,0), radius = 20, color = color.cyan)
    if i==3:    #add fourth mass to the bottom of the third one
        addedmass3 = cylinder(pos = mass.pos - (0,3*massHeight,0), axis = vector(0,-massHeight,0), radius = 20, color = color.orange)
        addedmass.pos +=(0,-delstretch,0) #lower the second mass
        addedmass2.pos +=(0,-delstretch,0) #lower the third mass
    unstretched_line = cylinder(pos=mass.pos - (-25,0,0), axis = vector(25,0,0), radius = 1, color = color.white)

    Fvsx.plot(pos = (stretch/scalefactor, m*g), color = color.red) #plot height vs. position graph
    print stretch/scalefactor, m*g
    scene.waitfor('click')

m = 0  #reset mass for the second spring

newmass = cylinder (pos = newspring.pos - (0,50,0), axis = vector(0,-massHeight,0), radius = 20, color = color.yellow)

#repeat for the second spring
for i in range(4):
    m +=delm
    delstretch = scalefactor* delm*g/k2
    newstretch = scalefactor* (m*g/k2)  #stretch of the spring
    newspring.length += delstretch  #calculate the new length of the spring
    newmass.pos +=(0,-delstretch,0)
    #newmass.length += 5   from when the cylinder just stayed the same color but got longer
    if i==1:  # add the second mass to the bottom of the first one
        newaddedmass = cylinder(pos = newmass.pos - (0,massHeight,0), axis = vector(0,-massHeight,0), radius = 20, color = color.red)
    if i==2:  #add third mass to the bottom of the second one
        newaddedmass.pos +=(0,-delstretch,0)  #move second mass to new, lower position
        newaddedmass2 = cylinder(pos = newmass.pos - (0,2*massHeight,0), axis = vector(0,-massHeight,0), radius = 20, color = color.cyan)
    if i==3:    #add fourth mass to the bottom of the third one
        newaddedmass3 = cylinder(pos = newmass.pos - (0,3*massHeight,0), axis = vector(0,-massHeight,0), radius = 20, color = color.orange)
        newaddedmass.pos +=(0,-delstretch,0) #lower the second mass
        newaddedmass2.pos +=(0,-delstretch,0) #lower the third mass


  #  newmass.length += 5
    Fvsx.plot(pos = (newstretch/scalefactor, m*g), color = (0,0.7,0)) #plot height vs. position graph
  #  print newstretch/scalefactor, m*g
    unstretched_line = cylinder(pos=newmass.pos - (-25,0,0), axis = (25,0,0), radius = 1, color = color.white)
    scene.waitfor('click')

scene.waitfor('click')
exit()
```

CS5 for All:

An Introduction to Computer Science and Python Programming

ID VERIFIED

CERTIFICATE of ACHIEVEMENT

edX HARVEY MUDD COLLEGE

Zachary Dodds
Professor Department of Computer Science
Harvey Mudd College

This is to certify that

Phillip Wolf

successfully completed and received a passing grade in

CS005x: CS For All: Introduction to Computer Science and Python Programming

a course of study offered by HarveyMuddX, an online learning initiative of Harvey Mudd College through edX.

VERIFIED CERTIFICATE
Issued September 18, 2015

Verify the authenticity of this certificate at
https://verify.edx.org/cert/3dc422c8004f4a73891ec5e337cbaf6f

CS5 for All

From July through late September, 2015, I took the Computer Science course *CS5 Computer Science for All*, from Harvey Mudd College, through edx.org, a consortium of Universities that offer their classes online for free, or you can pay $49 to get a certificate when you complete the course.

This particular course advertises that you'll learn:

- Basic Python Programming
- Design, implementation, documentation, and testing skills
- Strategies for solving computational problems
- Applications of CS in society and real world context

and that the expected weekly effort is 5-7 hours per week for 14 weeks.

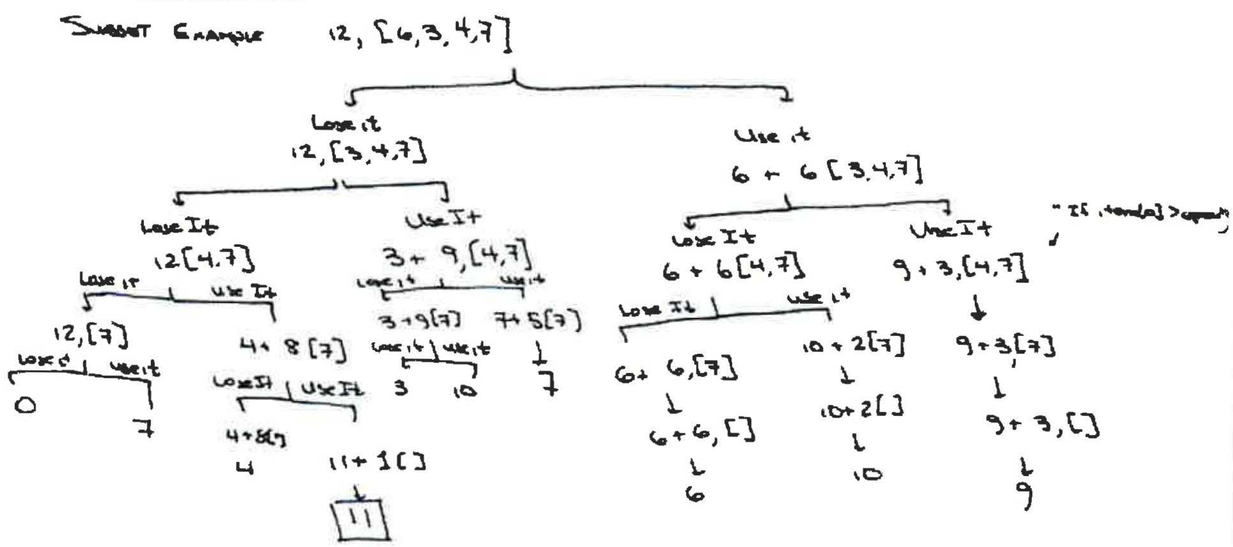
My Experience of the Course

This was the first online course I have taken through edX.This course easily takes at least three times as much time as they suggested it would. Some of this is the nature of computer programming courses—a *piece* of a problem that you might bang your head against for a day and a half can end up sometimes being solved with two or three lines of code. A program twenty lines long can sometimes take days to get to work, because you have to think like the computer does.

The programming puzzles posed in the course were generally challenging, sometimes frustrating, and usually fun in the sense that they required creative thinking. The most challenging part was dealing with recursion, in which you have a procedure that calls itself, each time going deeper into a stack of procedure calls until you final meet some terminating condition, after which the computer works its way out of the procedures again starting from the bottom an ending with the original procedure call. It is a slow, memory-intensive way to get things done. Some of the problems were exercises where using a recursive procedure was required even through there were quicker and easier ways to do things.

One of the most clever and difficult programs to write was one called "subset". It takes a given target number (say, 12) and a set of other numbers (say, [6,3,4,7]) and comes up with the sum of numbers in that set that gets as close as possible to the target number. In this particular case you can just look at it and see that 4 + 7 = 11 is as close as you can get. But this clever program***, in just 8 lines***, can evaluate every possibility by trying the sum by keeping the first number "6" and then applying the process to what's left in the set [3,4,7] and trying to get a match to 12-6 = 6 (**useIt**). Or, the procedure can dump the first number "6" (**loseIt**) and apply the process again to what is left in the set [3,4,7]. The procedure keeps calling itself, over and over again, until it has exhausted every element in the set [6,3,4,7] and evaluated every combination. This is an example of a program that looks short and easy, but which takes days to figure out! Fun!

```
def subset(capacity,items):
    """
    Given a maximum capacity and a list of positive numbers,
    choose a set of numbers whose sum is closest to the maximum
    capacity
    """
    if capacity <= 0 or items == []:
      return 0
    elif items[0]>capacity:
      return subset(capacity,items[1:])
    else:
      loseIt=subset(capacity,items[1:])
      useIt=items[0]+subset(capacity-items[0],items[1:])
      return max(useIt,loseIt)

def exact_change( target_amount, L ):
    """
    the input target_amount is a single non-negative integer
    value and the input L is a list of positive integer values.
    Function returns True if you can make the target amount exactly
    from the integers in the list. Returns False if you cannot
    """
    if subset(target_amount,L) == target_amount: return True
    else: return False
```



Every edX course has an online discussion list. They also have at least one online TA, who scans through the forum posts a couple of times each day and responds to questions posed by students in the course. That particular feature was invaluable here in that the online course was relatively new, the online program grading feature was at times buggy, and the assignment directions were at times unclear.

I achieved my goal of learning how to program in Python. I also learned a lot of other things about computer science and programming in general, much more than I needed to fulfill my sabbatical project. In retrospect it would have been more "efficient" to take a dedicated Python programming course rather than a Harvey Mudd Computer science course.

Here's a summary of I learned/studied/did as projects:

Lesson 1:

Program the computer to play rock paper scissors

--ask for input from the player

--computer randomly chooses rock, paper, or scissors, then determines if you win, lose, or tie

Program in a language called Picobot

--you are on a square grid. You give the picobot a list of instructions specifying:

Its state (given by a number)

A set of conditions (is North, East, West or South free, blocked, or do you care)

Which way to move if that set of conditions is met)

What state to set the picobot 2

A "program" consists of a set of instructions, in order, where the computer checks each set of instructions in order from the top down, and executes the first one that is True. For example, if the program line read 0 NxW* ->E2 this means

"If the bot is in state 0, North is blocked, East is free, West is blocked, and you don't care about South, then move one square to the East on the grid and reset your state to 2"

If these conditions aren't met, the program goes to the next line to see if those conditions are met. If so, the program executes that step and then goes back to the top. If not, it goes to the next one, and so on.

What was required was a program to fill an 20 x 20 space, a 20 x 20 space that had some obstacles in it, and then to solve a maze by covering every possible square in it. Here's an example of a maze solving program that follows the "right hand rule"—if you can go right, do so. If you can't go straight, if you can't, go left, and if you can't do that, go backwards.

Here is the program that worked:

```
# Phillip Wolf
# July 7, 2015
# This Picobot program will fill a maze

# state 0 with nothing N: go one step N
0 x*** -> N 0
# state 0, only to the N: go E + into st 1
0 Nxxx -> E 1
0 NExx -> W 2
0 NxWx -> E 1
0 NEWx -> S 3

#state 1 with nothing E; go East
1 *x** -> E 1
1 NExx -> S 3
```

1 xExS -> N 0
1 NExS -> W 2

#state 2 with nothing W; go West
2 **x* -> W 2
2 NxWx -> S 3
2 xxWS -> N 0
2 NxWS -> E 1

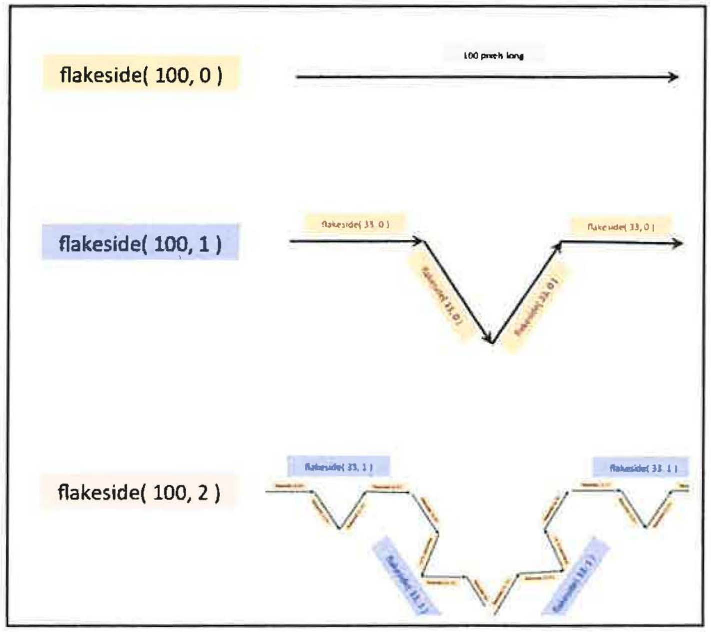
#state 3 with nothing S; go South
3 ***x -> S 3
3 *ExS -> W 2
3 *xWS -> E 1
3 xEWS -> N 0

Lesson 2:
-Learning how to slice and index a list in Python—how to pick out, say, the 8th element in a list, or just the odd elements.
Wrote a series of procedures to
--take two numbers and find the square, or the midpoint, or some geometric mean.
-Determine if the first and last letter in a string (series of letters) are the same.
-- inputs a string. Switches the first and second half of a string.
If the string has an odd number of characters, the first half has
one fewer characters than the second half.
--inputs a number of seconds. Formats output as day, hours, min, sec
--Given a string, we'll say that the front is the
first 3 chars of the string.If the string length is less
than 3, the front is whatever is there.
Return a new string which is 3 copies of the front.

Lesson 3: Recursion

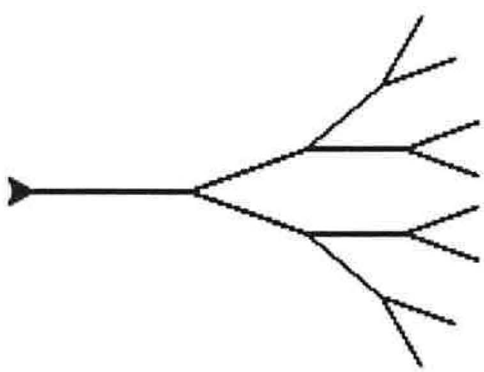
Introduces turtle graphics, which lets you draw lines of various lengths in various directions. Here is an example of a procedure ("flakeside")

```
# python 2
# Name: Phillip Wolf
# Date: July 12, 2015
from turtle import *
def flakeside( sidelength, levels):
  """ Draws one side of a snowflake"""
  if (levels<1):
    forward (sidelength)
    return
  else:
    flakeside(sidelength/3, levels-1)
    right(60)
    flakeside(sidelength/3, levels-1)
    left(120)
    flakeside(sidelength/3, levels-1)
    right(60)
    flakeside(sidelength/3, levels-1)
    return
#
```



What makes this interesting (and frustrating!) is that the procedure can call itself halfway through, so you have to keep track of how many levels deep you are into the procedure to figure out what it is going to do next.

I also figured out how to draw progressively more detailed snowflakes, or square spirals where each line is some percent longer or shorter than the previous out, and changes color as we go, or a program that draws a tree using a self-referential branching structure:



This took me days to get right.

There were also procedures to
--take the dot product of two vectors
--determine the scrabble score of any word
--convert any DNA sequence and return the corresponding RNA sequence
--Do a random walk, where a "sleepwalker" starts somewhere on a number line, and can go either to the right or left each step, and you determine where the sleepwalker is after *n* steps
--As a variation, start somewhere, set possible ending points to the right and left, and determine how many steps it takes a sleepwalker until he gets to one end or the other.
--Find the ending position of a sleepwalker after 100 steps.
--Do the above for *n* trials, and calculates the average distance and root-mean-square distance that the sleepwalker ends up away from the starting point after 100 steps
--divide a numerical interval into *n* equally spaced steps
--divide a numerical interval into *n* equally spaced steps, then applies some function *f* to each number along the way
--numerically integrats a function *f* by breaking the input interval into *n* steps of equal width Δx and then calculating the area under the graph of the function by summing the area of the rectangles of height *f* times the subinterval width Δx and adding them up over the entire interval

Lesson 4

Problem 1
--develop a set of procedures that takes element *i* in a list and changes it in some way (add 1, square, multiply by 2, etc)., and returns a new list with that changed element
--develop a procedure that take the last element of a list and moves it to the beginning, repeating this by some specified number of times.
--randomly replaces element *i* in a list with either a 0 or a 1
--returns the value True if every element in a list is a one.
--changes only element *i* in a list to a 1
--toggles element *i* in a list (so if it was zero, now it's 1, and vice versa)
--changes only element *i* in a list to a 1 and also toggles the elements before and after it in a list
--returns a random binary list of 1s and 0s
--write a "game" where the computer generates a random binary list and randomly changes one element (and toggles its neighbors) until the list ends up as a list of all 1s.

Problem 2
--Generates a scrabble score for a word
--takes a letter of the alphabet and shifts it *n* steps up the alphabet (distinguishes between lower case and upper case too!, if you get to z the next steps is back to a)
--enciphers a string of text by shifting the entire string up the alphabet by *n* steps
--decodes a shifted string by generating all possible shifted strings (shifted by 1-25 steps up the alphabet, evaluates the scrabble score for each string, and guesses that the best string is the one with the lowest scrabble score
--takes a list of ones and zeroes and sorts it into zeroes first, then ones
--find the maximum value in a list and moves it to the end from wherever it was
--takes a list and sorts it in order low to high
--defines a jotto score by comparing two words and returning the number of characters that are the same in both words

--given a list of numbers and a sum that you are trying to reach, figures out which numbers in the list will get you closes to the sum

{This is a really clever recursive problems that follows a strategy called "use it or lose it". It calculates the sum with or without that number. The procedure keeps a running total of all possibilities by calling itself over and over again.

--write an exact change procedure that determines if you can get exactly to a desired total using some or all of the numbers in a list

--write a procedure that takes two strings (say, S and T) and finds the longest sting of characters that they have in common AND that are in order. So if you compare "human" and "chimp" the program returns "hm"

Lesson 5

Write procedures that

--determine if a number is odd or even

--convert a number to binary

--convert a binary string to a base 10 number

--take an 8-bit binary string, add 1 to it, and then output the new 8-bit string (no carrying)

--takes an 8-character binary string and counts upward from there *n* steps going one step at a time, printing out each step as it goes

--converts a decimal number to a ternary (base three) string

--Takes in a balanced ternary number

(instead of 0,1,2 you use +, -, and 0 in each "place" in base three, so + 0 – means (1(9)+0(3)- 1(1) = 8)

--convert a decimal number to a number in base B

--convert a number in base B to a decimal number

--convert a number in base B1 to a number in base B2

--add two binary strings and returns their result as another binary string

--sums two binary digits and returns a sum value for that digit (0+0 = 0, 1+0 =1, 0+1=1, 1+1=0)

--sums two binary digis and returns the "carry" value (so 1+1 returns 1; returns zero otherwise)

--removes the leading zeroes from a string. So 0001101 becomes 1101

--adds two binary strings. The approach is to

Make the strings the same length.

Add them and create two separate strings, one to keep track of the sum, and the other to keep track of the carry string.

Then add these two strings by calling the function again.

--convert a decimal number to a binary and then add enough leading zeroes to make it 7 bits long

--so, 17 becomes 100001 becomes 0100001

--make compression and decompression procedures that takes a binary string and then makes a list of 8-bit numbers. The first digit of this 8-bit number tells you whether the digit in the original number is a zero or a one, and the next seven digits tell you how many ones or zeros there are in a row. Decompress takes you back from this list to your original string.

Lesson 6—Logic gates
--Use a legacy program called Logisim to build logic circuits from logic gates
--Build an XOR gate from AND, OR and NOT gates
--Build a "full adder" that takes two digits and the carry term from a previous addition and outputs the sum
--Build a ripple carry adder by using four full adders. This takes in two four bit numbers and outputs a sum that could be five bits long
--Build a four bit multiplier in Logisim, which will take two four-bit binary numbers and multiply them to get an 8-bit number.
--Build an integer division circuit that divines a 3 bit number by a two bit number to give a result. (Integer division leaves off remainders)

Lesson 7: Assembly language
This gets into the nuts and bolts of how computers manage memory by moving numbers around in registers and doing various operations on them. They syntax is non intuitive!
--program a pseudo random number generator in assembly, then check the results in Python
--write an assembly program that takes two non-negative numbers, raises the first number to the power of the second number, and then outputs the result.
--write a program that gets a number input from the user, then generates that many terms of the Fibonacci sequence.
--modify a program that calculates factorials to calculate powers of a number instead

Lesson 8: Loops in Python
Problem 1
--takes a number and raises it to a power by multiplying it by itself the appropriate number of times in a FOR loop
--takes a list of integers and sums just the odd numbers in the list
--takes a number between 0 and 99. Has the computer "guess" the number repeatedly until it gets the correct guess. Counts the number of guesses required
--generates a random list of integers between zero and some maximum value, then goes through the list counting until it comes to a repeated value

Problem 2
Write procedures to
--generate a random x, y coordinate with x and y each between -1 and +1, then another procedure to determine whether that coordinate is within a circle of radius 1 centered on the origin, or not. Then calculate pi as 4* the fraction of generated coordinates that lie in the circle
--write a procedure which takes in an acceptable error using this method and determines how many coordinates it has to generate to get a value of pi within that acceptable error.

Problem 3
Write a series of procedures to input a series of day to day stock prices and then calculate various quantities based on the contents of the list

Lesson 9

PROBLEM 1: THE GAME OF LIFE

The Game of Life is a *cellular automaton* invented by John Conway, a mathematician from Cambridge. The game of life is not so much a "game" in the traditional sense, but rather a process that transitions over time according to a few simple rules. The process is set up as a grid of cells, each of which is "alive" or "dead" at a given point in time. At each time step, the cells live or die according to the following rules:

1. A cell that has fewer than two live neighbors dies (because of isolation)
2. A cell that has more than 3 live neighbors dies (because of overcrowding)
3. A cell that is dead and has exactly 3 live neighbors comes to life
4. All other cells maintain their state

Although these rules seem simple, they give rise to complex and interesting patterns. For more information and a number of interesting patterns see the Wikipedia article on the Game of Life.

In this lab, you will implement a Python program to run the Game of Life.

THINKING ABOUT LIFE

As always, it is important to break the problem down into pieces and develop the program in stages so that others can understand the code and so that you can ensure that each piece is correct before building on top of it. We will break this problem down into the following steps:

- Creating a 2d array of cells
- Displaying the board and updating it with new data
- Allowing the user to change the state of the cells
- Implementing the update rules for the "Game of Life"

Specifically, write procedures that:
--creates a row of zeros of width "width"
--creates a 2 dimensional array of height "height" and width "width
--prints out that array
--creates an empty board and then creates a diagonal strip of "on" cells in it (1s)
--prints an array with zeros on the boarders an ones inside
--returns an array of random ones and zero with zeros on the border
--makes a "deep copy" of an array so that future changes in the array aren't mirrored in the copy
--takes an array, reverses all of the inner contents but leaves the border cells alone
--returns the number of "live" neighboring cells for a given cell on the board
--advances the game of Life by one step, based on the rules

Problem 2 Tic Tac Toe + N
Write procedures to
--make a 2-dimensional array of NR rows and NC columns
--check if there are three Xs or three Os in a row, either horizontally, vertically, or diagonally
--check if there are *N* Xs or three Os in a row, either horizontally, vertically, or diagonally

Lesson 10

Problem 1

Learn about *methods*, which are built in functions or processes

--take any calendar date and advance it by one day

--take any calendar date and advance it by *N* days

--take any calendar date and move it back by one day

--take any calendar date and move it back by *N* days

--determine if one date is before or after another date, or if two dates are the same

--determines what day the first day of any decade occurred on

--given any calendar date, determines what day of the week it was (or will be)

Problem 2

write a program to play Connect Four, in which players stack Xs or Os, one at a time in a series of 7 columns, to try to get four in a row horizontally, vertically, or diagonally

--create a 7 x 6 board, then look for four in a row in any direction

--add an X or O to the board in a particular spot

--clear the board

--set up a board to check out the algorithms to see if someone has one or not

--determine if a move is legal

--determine if the board is full

--remove a move

--prompt a player for input, then determine if the move is legal or not

--check if someone has won

--host a full game—create the board, add moves one at a time, etc.

Problem 3—Markov Text generation

Here is the problem description:

This problem uses a Python dictionaries to model—and then generate—text. Python examples of how dictionaries can be used to analyze text in a vocabulary-counter (as opposed to a word-counter) and to keep track of guesses in a guessing-game are available in this trinket.

Markov Text Generation

Here's the basic idea: English is a language with a lot of structure. Words have a tendency (indeed, an obligation) to appear only in certain sequences. Grammatical rules specify legal combinations of different parts of speech. E.g., the phrase "The cat climbs the stairs" obeys a legal word sequence. "Stairs the the climbs cat", does not. Additionally, semantics (the meaning of a word or sentence), further limits possible word combinations. "The stairs climb the cat" is a perfectly legal sentence, but it doesn't make much sense and you are very unlikely to encounter this word ordering in practice.

Even without knowing the formal rules of English, or the meaning of English words, we can get idea of what word combinations are legal simply by looking at well-formed English text and noting the combinations of words that tend to occur in practice. Then, based on our observations, we could generate new *sentences by randomly selecting words according to commonly occurring sequences of these words. For example, consider the following text:*

"I love roses and carnations. I hope I get roses for my birthday."

If we start by selecting the word "I", we notice that "I" may be followed by "love", "hope" and "get" with equal probability in this text. We randomly select one of these words to add to our sentence, e.g. "I get". We can repeat this process with the word "get", necessarily selecting the word "roses" as the next word. Continuing this process could yield the phrase "I get roses and carnations". Note that this is a valid English sentence, but not one that we have seen before. Other novel sentences we might have generated include "I love roses for my birthday," and "I get roses for my birthday".

More formally, the process we use to generate these sentences is called a ***first order Markov process*** *. A first-order Markov process is a process in which the state at time* t+1 *(i.e. the next word) depends only on the states at times* t *(i.e., the previous word). In a second-order Markov process, the next word would depend on the* two *previous words, and so on. Our example above was a first order process because the choice of the next word depended only on the current word. Note that the value of the next word is independent of that word's position and depends only on its immediate history. That is, it doesn't matter if we are choosing the 2nd word or the 92nd. All that matters is what the 1st or the 91st word is, respectively.*

Your Text Analyzer and Generator

In the first part of this assignment you will implement a first-order Markov text generator. Writing this generator will involve two functions:

1. *One to process a file and create a dictionary of legal word transitions and*
2. *Another to actually generate the new text.*

--Takes a list of words and makes a list where each element is a list consisting of that single word

--Takes a txt file.
takes a List L, where each element is a string (a word from a text).
Then converts this to a list LoW where each word is its own list,
then populates a dictionary as follows.
1) Every word that starts a sentence becomes a value for the key '$'
2) Every other word becomes a value for the word before it.
We end up with a dictionary with a set of keys, with possibly multiple values assigned to each key.

-- Takes in a dictionary of word transitions d (generated in the createDictionary function, above) and a positive integer, n. Then, generateText should return a string of n words.
1) The first word should be randomly chosen from among those that can follow the sentence-starting string "$".
2) The second word will be randomly chosen among the list of words that could possible follow the first, and so on.
3) When a chosen word ends in a period ., a question mark ?, or an exclamation point !, the generateText function should detect this and start a new sentence by again choosing a random word from among those that follow "$".

--Finally, import some well structured English text from some document, then use the program to generate some "meaningful" text using the English structural patterns from the first document.

Here is the original text file I used to generate the dictionary for the Markov essay:

Spirituality is inefficient.
Organized religion tries to package spirituality into efficient one-hour experiences.
While science slowly provides answers to life's persistent questions, religion is an anchor binding us to a past of former mystery, superstition and self-proclaimed authority.
I have chosen articles for this section from writers who agree with me, such as:
"I'm certain that magicians of the past would have readily appropriated many of the methods and discoveries of science along with the accompanying technologies. The tendency of reason and science to take up too much room in modern life is just another symptom of disenchantment. The root problem is not science. It is religion."
Some contemporary observers have argued, echoing generations of religious apologists, that the resurgence of religious expression testifies to the spiritual sterility of technological rationality, that religious belief is now being renewed as a necessary complement to instrumental reason because it provides the spiritual sustenance that technology lacks. There is perhaps some truth to this proposition, but it still presupposes the mistaken assumption of a basic opposition between these two phenomena and ignores what they have in common. For modern technology and modern faith are neither complements nor opposites, nor do they represent succeeding stages of human development. They are merged, and always have been, the technological enterprise being, at the same time, an essentially religious endeavor.
This is not meant in a merely metaphorical sense, to suggest that technology is similar to religion in that it evokes religious emotions of omnipotence, devotion, and awe, or that it has become a new (secular) religion in and of itself, with its own clerical waste, arcane rituals, and articles of faith.
Rather, it is meant literally and historically, to indicate that modern technology and religion have evolved together and that, as a result, the technological enterprise has been and remains suffused with religious belief.
The expectation of ultimate salvation through technology, whatever the immediate human and social costs, has become the unspoken orthodoxy, reinforced by a market-induced enthusiasm for novelty and sanctioned by a millenarian yearning for new beginnings. This popular faith, subliminally indulged and intensified by corporate, government, and media pitchmen, inspires an awed deference to the practitioners and their promises of deliverance while diverting attention from more urgent concerns. Thus, unrestrained technological development is allowed to proceed apace, without serious scrutiny or oversight—without reason. Pleas for some rationality, for reflection about pace and purpose, for sober assessment of costs and benefits—for evidence even of economic value, much less larger social gains—are dismissed as irrational. From within the faith, any and all criticism appears irrelevent, and irreverent.

Here is the 500-word essay generated by the program:

From within the accompanying technologies. Organized religion is not science.

The root problem is now being renewed as a market-induced enthusiasm for novelty and modern technology lacks. From within the spiritual sterility of the immediate human and discoveries of ultimate salvation through technology, whatever the accompanying technologies. This is allowed to a necessary complement to package spirituality into efficient one-hour experiences. The tendency of economic value, much room in a result, the practitioners and articles for novelty and discoveries of faith. I have readily appropriated many of omnipotence, devotion, and irreverent. Thus, unrestrained technological rationality, for new beginnings. The expectation of costs and articles for sometruth to religion in modern technology lacks. For modern faith are neither complements nor opposites, nor opposites, nor opposites, nor do they have evolved together and benefits--for evidence even of reason and discoveries of disenchantment. Pleas for new (secular) religion in modern life is not meant literally and always have chosen articles for this proposition, but it is now being renewed as a new beginnings. While science along with religious belief is now being renewed as a merely metaphorical sense, to suggest that modern technology and religion is perhaps some rationality, for sober assessment of the unspoken orthodoxy, reinforced by a result, the faith, any and of costs and sanctioned by corporate, government, and social costs, has been and always have argued, echoing generations of religious emotions of science along with the immediate human development. The tendency of technological rationality, for some truth to the faith, any and media pitchmen, inspires an essentially religious belief is allowed to this proposition, but it is not meant in modern faith are merged, and historically, to the technological rationality, for new (secular) religion in that magicians of ultimate salvation through technology, whatever the same time, an anchor binding us to life's persistent questions, religion in that magicians of former mystery, superstition and of a necessary complement to indicate that the same time, an essentially religious endeavor. There is an anchor binding us to suggest that technology is not meant in modern life is similar to a result, the mistaken assumption of the resurgence of religious expression testifies to the methods and their promises of itself, with the spiritual sterility of deliverance while diverting attention from more urgent concerns. Rather, it provides the methods and of former mystery, superstition and religion tries to life's persistent questions, religion tries to a past of religious endeavor. While science to suggest that the unspoken orthodoxy, reinforced by a result, the mistaken assumption of religious emotions of economic value, much room in that magicians of a merely metaphorical sense, to this section from writers who agree with religious belief. Pleas for new (secular) religion in that technology and purpose, for reflection about pace and of itself, with its own clerical waste, arcane rituals, and that, as irrational. For modern technology and their promises of faith. Pleas for reflection about pace and awe, or that religious apologists, that magicians of technological development is

Circuits and Electronics:

Basic Circuit Analysis

VERIFIED

CERTIFICATE of ACHIEVEMENT

MITx

This is to certify that

Phillip Wolf

successfully completed and received a passing grade in

6.002.1x: Circuits and Electronics 1: Basic Circuit Analysis

a course of study offered by MITx, an online learning initiative of the Massachusetts Institute of Technology through edX.

Anant Agarwal
Professor, Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Bonnie Lam
Graduate Instructor, Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Sanjay Sarma
Dean of Digital Learning
Massachusetts Institute of Technology

edX

VERIFIED CERTIFICATE
Issued January 28, 2016

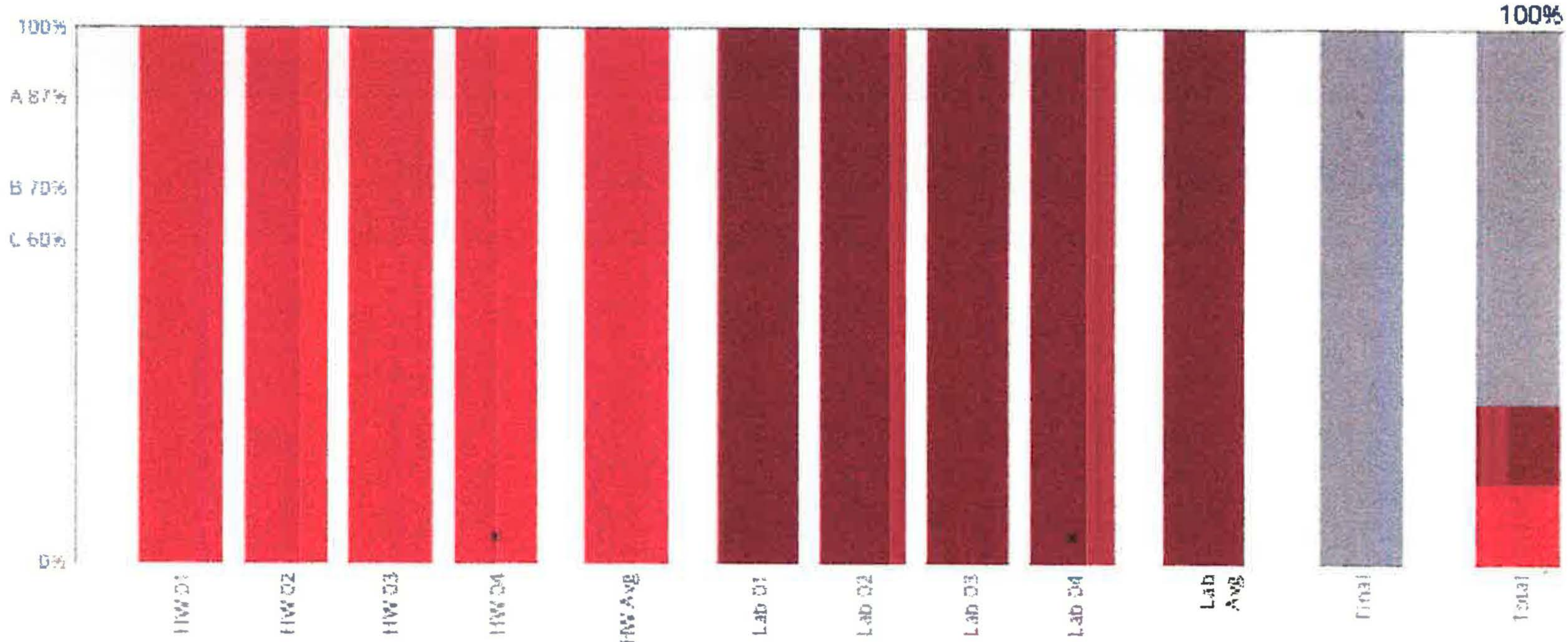
VALID CERTIFICATE ID
02525c03987640c2b9c1494533024620

Course Progress for Student 'PhilWolf' (phil.wolf@sbcglobal.net)

Your certificate is available

You can keep working for a higher grade, or request your certificate now.

View Certificate



MITx 6.002.1x
Circuits and Electronics 1: Basic Circuit Analysis

Syllabus

Week* 1

Topics	Lumped circuit abstraction, circuit elements, KVL, KCL, simplification techniques, nodal analysis
Readings**	1, 2.1-2.5, 3.1-3.5
Graded assignments	HW1, Lab1

Week 2

Topics	Linearity, superposition, Thevenin & Norton methods, digital abstraction, Boolean logic, combinational gates
Readings	3.5-3.6, 5.1-5.4, 5.6-5.7
Graded assignments	HW2, Lab2

Week 3

Topics	MOSFET switch, MOSFET switch models, nonlinear resistors, nonlinear networks
Readings	6.1-6.8, 4.1-4.3
Graded assignments	HW3, Lab3

Week 4

Topics	Small signal analysis, small signal circuit model, dependent sources, analog amplification
Readings	4.5, 2.6, 7.1-7.2
Graded assignments	HW4, Lab4

Final Exam

* The term "Week" is used to indicate the length of time allocated for the topics listed in the original 6.002x course. It is also the pace at which the course is taught at MIT. Since this course is self-paced, you may choose to allocate more or less time to study the materials. The suggested workload for this course is approximately 6 hours per "week".

** Readings refer to sections in the course textbook.

The reality is that this course took about three times the estimated 24 hours they suggested!

Week 1:
Kirchoff's Voltage Loop and Current Loop methods—basically V = IR and sum of currents at a node equals zero.
Node method—Label the voltages at each node. Write the current node equations in terms of voltage differences and conductivities.
Model batteries as ideal voltage sources with internal resistances
Current voltage and power calculations for resistive circuits
Simplification of resistances using parallel and series rules
Constant voltages sources and constant current sources in circuits

Week 2:
Linear combinations of source strengths
superposition method of solving circuits
Thevenin approach to solving circuits
Norton method of solving circuits
Logic gates and truth tables
Current dividers
Modeling solar panels as a constant current source with some internal resistance in parallel and series to the source
"LAB": constructing a circuit that combines to output signals

Week 3:
Using MOSFETs as switches
Using "switches" to build logic gates, taking into account that the gates don't switch at a specific value, but might switch over a range of voltage values
current-voltage characteristics of Zerner diodes, diodes
Modeling a circuit with a non-linear element

Week 4:
Linearization—how to use a non-near element in a circuit in such a way that it gives a linear response to an input
Using dependent voltage sources and dependent current sources in circuits, and calculating Thevenin and Norton equivalents

The course is broken up into "weeks". Each week consists of a LOT of short video lecture segments, each a few minutes long. Almost always the lecture consisted of a set of powerpoint slides, often with "animations", with a voiceover and occasional scribbles on the screen. Often a segment would end with a "Here's a question to ponder. Think about it for a minute, and I'll see you in the next segment." I found this approach particularly useful, and this was the approach I ultimately decided to adopt in putting together my own course.

All of the homework is numerical, with the questions posed with spaces for you to input your answers. The online system checks your work and gives you several tries to get the answer right. Rather than including pages and pages of what is mostly screen shots, I have included here just my work for the practice exam and final exam for the course.

PRACTICE EXAM

Q1

The power distribution network in a particular apparatus can be modeled by the circuit below.

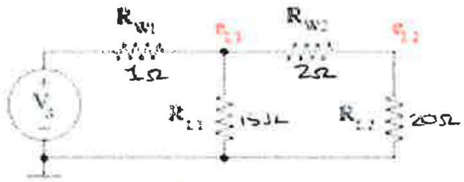


Figure 1-1

The power comes from a voltage source and is delivered to two loads. The loads are modeled by the resistors R_{L1} and R_{L2}. The resistances of the interconnects are modeled by the resistors R_{W1} and R_{W2}. The values of the element parameters are $V_S = 4.0\text{V}$, $R_{L1} = 15.0\Omega$, $R_{L2} = 20.0\Omega$, $R_{W1} = 1.0\Omega$, $R_{W2} = 2.0\Omega$.

Determine the node potentials e_{L1} and e_{L2}, assuming a ground node as indicated in the figure. Write your answers in the spaces provided below. Express your answers in Volts.

$e_{L1} =$

3.5967 ✔

$R_{eq} = 1 + \frac{15 \times 22}{37} = 9.19\Omega$

$I_{circ} = 4V / 9.19\Omega = 0.4033A$

$e_1 = 4 - .4033 = 3.5967\ V$

$e_{L2} =$

3.27 ✔

Determine the power dissipated in each of the resistors and the power entering the source. Express your powers in Watts. Remember that the power entering a two-terminal device is the product of the voltage across the device and the current through it in the associated reference directions. Write your answers in the spaces provided below.

$P_{R_{W1}} =$

0.1626 ✔

$P_{R_{W2}} =$

0.862 ✔

$P_{R_{L1}} =$

0.0535 ✔

$P_{R_{L2}} =$

0.535 ✔

$P_{V_S} =$

-1.613 ✔

	V	I	P
W1	(3) .4033	(2) .4033	.1627
[illegible]	[illegible]	.25	.35
L_1	4 - .4033 (15Ω) 3.5967 →	0.2397	0.862
L_2	3.27	.1635	.535
Source	4V	(1) 0.4033	1.613

1.613 W

Q3

The circuit below contains two dependent sources: a voltage controlled voltage source and a voltage controlled current source.

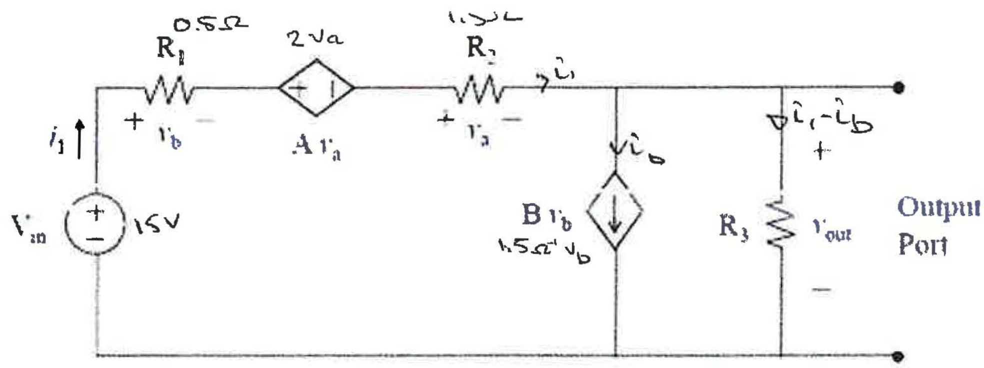


Figure 3-1

The circuit elements have the following values: $V_{in} = 15V$, $R_1 = 0.5\Omega$, $R_2 = 1.5\Omega$, $R_3 = 5\Omega$, $A = 2$ and $B = 1.5$.

$V - iR_1 - A i R_2 - i R_2 - (i - i_b) R_3 = 0$

$i_b = B i_1 R_1$

$V - i_1 (R_1 + AR_2 + R_2 + R_3 - B R_1 R_3) = 0$

$i = \frac{V}{R_1 + (A+1)R_2 + (1 - BR_1)R_3} = \frac{15V}{.5 + 3(1.5) + (1 - .75) \cdot 5}$

What is the value of the current i_1 (in Amps)?

2.4 ✔

What is the value of the output voltage v_{OUT} (in Volts)?

3 ✔

$i_b = B i_1 R_1 = 1.5(2.4)(0.5) = 1.8A$ $\quad i_3 = i_1 - i_b = 0.6A$

$V_3 = 0.6A \times 5\Omega = 3V$

We wish to create a Thevenin equivalent model of the above shown circuit as seen from its Output Port.

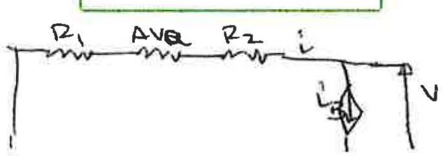
What is the value of the Thevenin equivalent voltage as seen from the Output Port (in Volts)?

3 ✔

3 V from above;

What is the value of the Thevenin equivalent resistance (in Ohms) as seen from the Output Port?

4 ✔



$V_b = -iR_1$

$V_a = -iR_2$

$V - iR_2 - A i R_2 - i R_1 = 0$

$-i(R_2 + AR_2 + R_1) = -V$

$i = \frac{V}{\ldots}$

$i_b = B V_b = -B i R_1$

$= (1 - BR_1)\left(\frac{V}{R_2 + AR_2 + R_1}\right)$

$\Rightarrow \frac{V}{I} = \frac{R_2 + AR_2 + R_1}{\ldots} = \frac{1.5 + 2(1.5)(0.5)}{\ldots} = \frac{5}{\ldots} = 20\Omega$

so $R_{TH} = 20 \,\|\, 5\Omega = R_3$ $\quad \frac{1}{20} + \frac{1}{5} = \frac{1}{4} \Rightarrow R_{TH} = 4\Omega$

Q3

The circuit below contains two dependent sources: a voltage controlled voltage source and a voltage controlled current source.

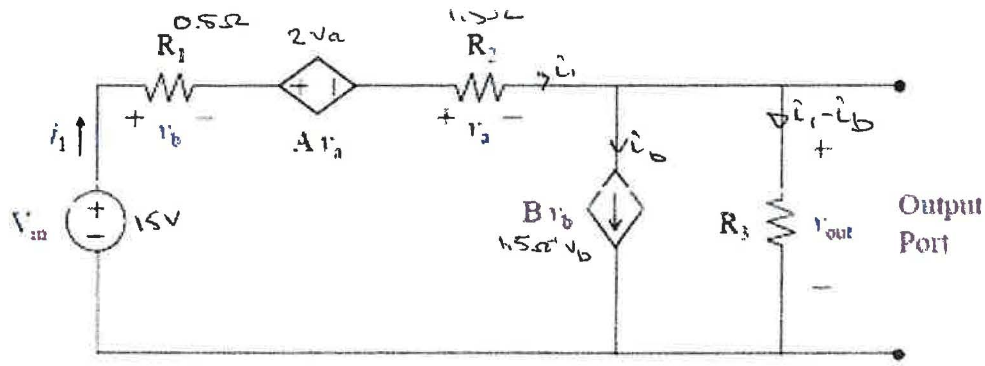


Figure 3-1

The circuit elements have the following values: $V_{in} = 15V$, $R_1 = 0.5\Omega$, $R_2 = 1.5\Omega$, $R_3 = 5\Omega$, $A = 2$ and $B = 1.5$.

$V - iR_1 - Ai R_2 - iR_2 - (i_1 - i_b)R_3 = 0$

$i_b = B i_1 R_1$

$V - i_1(R_1 + AR_2 + R_2 + R_3 - BR_1R_3) = 0$

$i = \frac{V}{R_1 + (A+1)R_2 + (1-BR_1)R_3} = \frac{15V}{.5 + 3(1.5) + (1-.75).5}$

What is the value of the current i_1 (in Amps)?

2.4 ✔

What is the value of the output voltage v_{OUT} (in Volts)?

3 ✔

$i_b = B i_1 R_1 = 1.5(2.4)(0.5) = 1.8A$ $\quad i_3 = i_1 - i_b = 0.6A$

$V_3 = 0.6A \times 5\Omega = 3V$

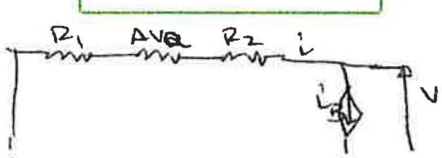
We wish to create a Thevenin equivalent model of the above shown circuit as seen from its Output Port.

What is the value of the Thevenin equivalent voltage as seen from the Output Port (in Volts)?

3 ✔ — 3V from above

What is the value of the Thevenin equivalent resistance (in Ohms) as seen from the Output Port?

4 ✔ — ignore R_3



$V_b = -iR_1$

$V_a = -iR_2$

$V - iR_2 - AiR_2 - iR_1 = 0$

$-i(R_2 + AR_2 + R_1) = -V$

$i = \frac{V}{\ldots}$

$i_b = BV_b = -BiR_1$

$= (1 - BR_1)\left(\frac{V}{R_2 + AR_2 + R_1}\right)$

$\Rightarrow \frac{V}{I} = \frac{R_2 + AR_2 + R_1}{\ldots} = \frac{1.5 + 2(1.5) + (0.5)}{\ldots} = \frac{5}{\ldots} = 20\Omega$

so $R_{TH} = 20 \parallel 5\Omega = R_3$ $\quad \frac{1}{20} + \frac{1}{5} = \frac{1}{4} \Rightarrow R_{TH} = 4\Omega$

Q4

For many purposes of gate design, we can model a MOSFET used as a switch simply as as an ideal switch and an "on-state resistor" R_{ON}. This is the SR model.

Assuming this model for the MOSFET, consider the inverter in the figure. This inverter is intended to be used as an element in a logic family with NAND and NOR gates.

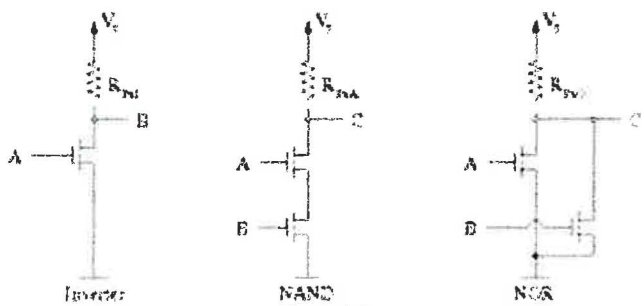


Figure 4-1

The static discipline required for this family is:

$$V_S = 5.0V, V_{OH} = 4.5V, V_{IH} = 4.0V, V_{IL} = 1.5V, V_{OL} = 1.0V.$$

What is the low noise margin (in Volts)?

0.5 ✔ $V_{IL} - V_{OL}$

What is the high noise margin (in Volts)?

0.5 ✔ $V_{OH} - V_{IH}$

What is the width of the forbidden region (in Volts)?

2.5 ✔ $V_{IH} - V_{IL}$

Suppose that the threshold voltage for the MOSFET is $V_T = 2.0V$ and $R_{ON} = 7000.0\Omega$.

What is the minimum value of the pullup resistor R_{PU} (in Ohms) for which this inverter can obey the required static discipline?

5V pulled down to 1V — Vs, 28K

28000 ✔

Now, consider the NAND gate of this family. What is the minimum value of the pullup resistor R_{PU} (in Ohms) for which this inverter can obey the required static discipline?

5V pulled down to 1V — Us, 56K

56000 ✔

How about the NOR gate of this family. What is the minimum value of the pullup resistor R_{PU} (in Ohms) for which this inverter can obey the required static discipline?

5V → 1V 1 gate open R=7000Ω — Vs, 28K, 7K

28000 ✔

Assume that we implemented this family with the minimum pullup resistors that you have already calculated.

What is the maximum power (in Watts) consumed by the inverter?

25/35000 ✔

$P = \frac{V^2}{R_{TOTAL}} = \frac{5V^2}{(2.8K + 7K)}$

What is the maximum power (in Watts) consumed by the NAND?

25/70000 ✔

$= \frac{5V^2}{(56K + 14K)}$

What is the maximum power (in Watts) consumed by the NOR?

25/31500 ✔

WITH BOTH GATES CLOSE

$P = \frac{V^2}{R} = \frac{5V^2}{(2.8K + 3.5K)}$

Q5

Consider the circuit in Figure 5-1 below.

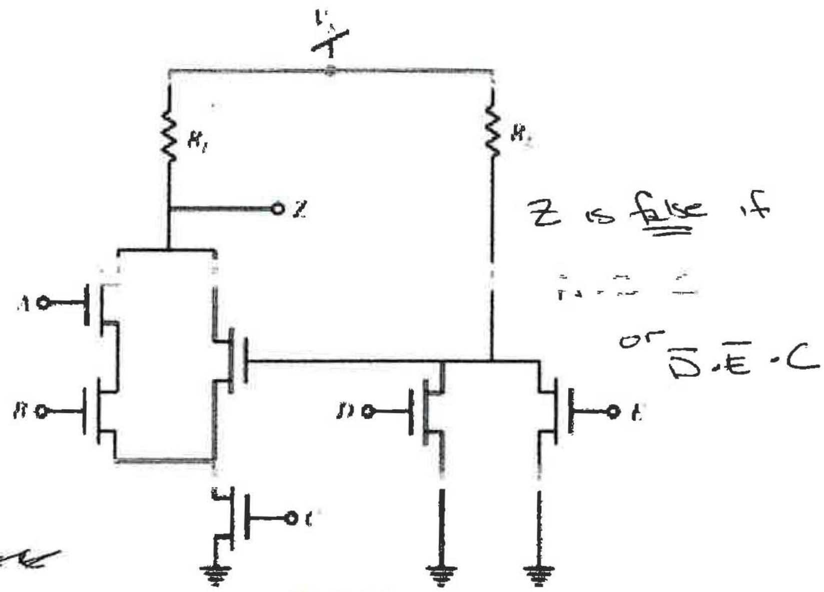


Figure 5-1

Write a boolean expression for Z in terms of $A, B, C, D,$ and E. You need not simplify your expression. Use your expression to fill out the truth table below.

A	B	C	D	E	Z	
1	0	1	0	1	z_0	not 0 = 1
1	1	1	1	1	z_1	0
0	0	1	1	0	z_2	not 0 = 1
0	1	0	0	0	z_3	0

Enter the unknown values for the outputs as one stream of bits in the form $z_0z_1z_2z_3$

1010 ✔

How many distinct boolean-valued functions are there of n boolean-valued signals? Write an expression in terms of n.

2^(2^n) ✔

N signals give 2^n possible combinations of inputs,

2^{2^n} distinct results

so 2 signals → 4 inputs 00 01 10 11

(?) they say...

Q6

A battery in combination with a dependent current source is connected to a lamp in the circuit shown below in Figure 6-1. The battery is modeled as a voltage source in series with a resistor, R_{bat}, and has an open circuit voltage of $4V$. The lamp turns on if the voltage across the lamp is greater than $2V$. When the lamp is on it has an internal resistance of 10Ω, and when it is off it acts like an open circuit.

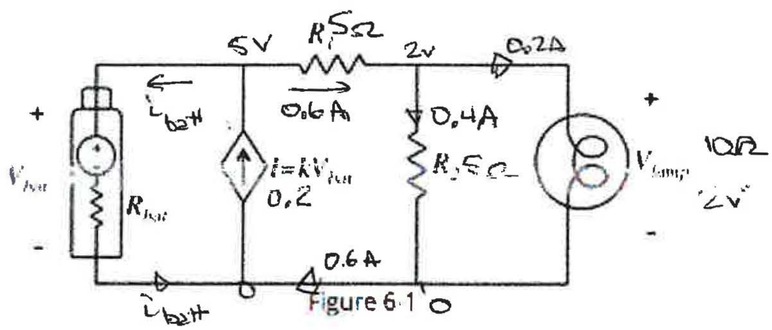


Figure 6-1

The elements in this circuit have the following values: $R_1 = 5\Omega$, $R_2 = 5\Omega$, and $k = 0.2$.

Assume that the lamp is on and $V_{lamp} = 2V$, what is the battery's internal resistance, R_{bat} (in Ohms)?

2.5 ✔

$V_{batt} = 5V = 4V + i_{batt} R_{batt}$

$I_{source} = 0.2 \times 5V = 1A$

$\rightarrow i_{batt} = 1 - 0.6A = 0.4A$

$i_{batt} R_{batt} = 1V \rightarrow R_{batt} = \frac{1}{0.4} = 2.5\Omega$

What is the power (in Watts) dissipated in the lamp?

0.4 ✔ $\frac{V^2}{R} = \frac{2^2}{10} = .4$

What is the power (in Watts) dissipated in R_1?

1.8 ✔ $\frac{3^2}{5} = 1.8\ \Omega$

What is the power (in Watts) dissipated in R_2?

0.8 ✔ $\frac{2^2}{5} = .8$

What is the power (in Watts) coming out of the voltage controlled current source, I?

5 ✔ $5V \times 1A$

What is the power (in Watts) coming out of the battery?

2 ✔ $-5V(0.4A)$

Q7

The circuit shown below is the linear equivalent model of a two-input single-output amplifier. Note that it contains a current dependent voltage source.

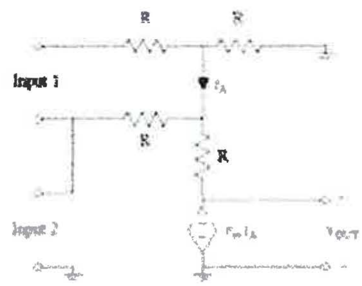


Figure 7-1

One application of this amplifier is in a communications circuit where its two inputs are driven by two antennas. We can model the two antennas as two current sources: i_{IN1} and i_{IN2}, as shown below.

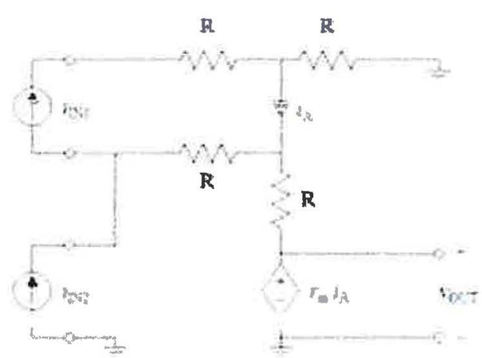


Figure 7-2

The elements in the circuit have the following values: $R = 3k\Omega$ and $r_m = 3k\Omega$

Assuming that $i_{N1} = 1mA$ and $i_{N2} = 0A$, what is the value of v_{OUT} in Volts?

2 ✔

Assuming that $i_{N1} = 0A$ and $i_{N2} = 1mA$, what is the value of v_{OUT} in Volts?

-1 ✔

Assuming that $i_{N1} = 1mA$ and $i_{N2} = 1mA$, what is the value of v_{OUT} in Volts?

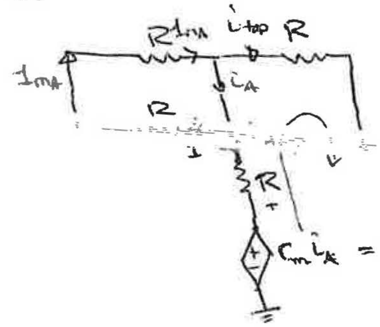
1 ✔

Assuming that input 2 is left as an open circuit, what is the Thevenin equivalent resistance (in kOhms) seen from input 1?

6 ✔

(calculations on next page)

7a



$i_A = 1\,mA - i_{top}$

[illegible] $- i_{top}R - i_{top}R = 0$

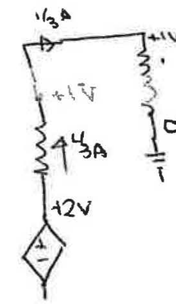
$r_m i_A = r_m(1 - i_{top})$

$r_m - i_{top}(r_m + R + R) = 0$

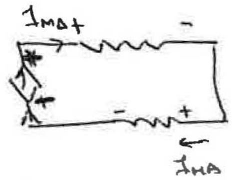
$i_{top} = \frac{r_m}{r_m + R + R} = \frac{1}{3}A$

$\rightarrow i_A = 1 - \frac{1}{3} = \frac{2}{3}A \Rightarrow V_{out} = \frac{2}{3}A \times$ [illegible]

Check

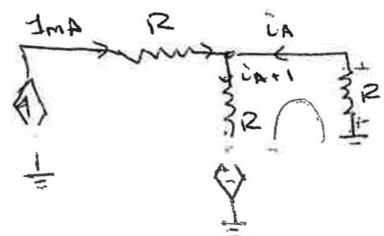


VOLTAGE ACROSS INPUT 1



$-1K + V_{input}$ [illegible] $= 0 \Rightarrow V_{input} = 2\,mA \cdot 3k\Omega = 6V$

7b



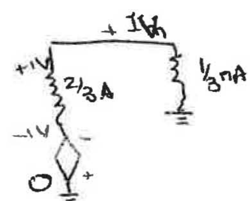
$-i_A R - (1 + i_A)R - i_A r_m = 0$

$-i_A($[illegible]$) - 1R = 0$

$i_A = \frac{-R}{2R + r_m} 1\,mA = -\frac{1}{3}\,mA$

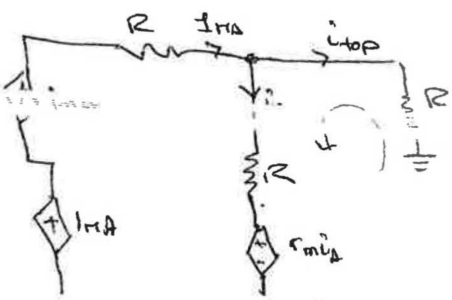
$\Rightarrow V_{out} = -\frac{1}{3}\,mA \times 3k\Omega = -1V$

Check



✓

7c



$i_A = 1 - i_{top}$

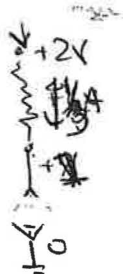
[illegible]

$i_{top}(2R + r_m) - 2R = 0$

$i_{top} = \frac{2}{3}\,mA$

[illegible] $\frac{1}{3}$ [illegible] $= 1V$

check



-2V

0

check.

7d

FINAL
EXAM

Final exam:

Q1 (10/10 points)

A linear circuit is shown below in Figure 1-1.

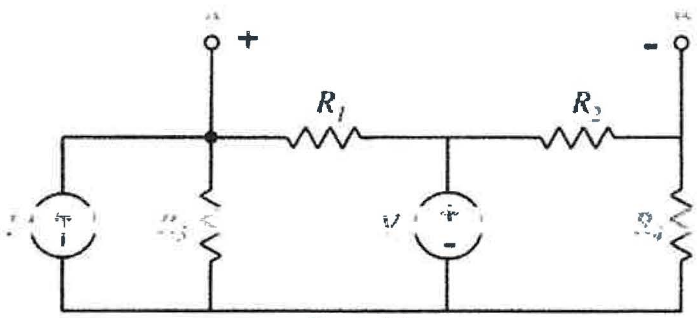


Figure 1-1

The elements in this circuit have the following values: $R_1 = 7.2k\Omega$, $R_2 = 12k\Omega$, $R_3 = 3.3k\Omega$, $R_4 = 4.7k\Omega$, $V = 5V$, and $I = 3mA$.

(a) Calculate the numerical value for the Norton equivalent current, I_N (in mA), for the A-B terminal. Express your answer to **two decimal places.**

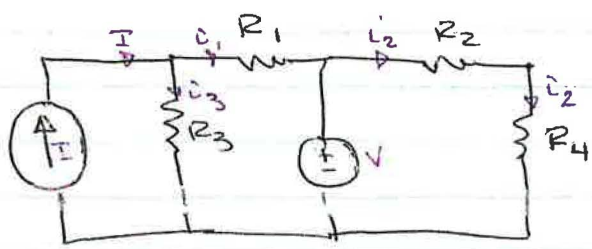
1.23 ✔

(b) Calculate the numerical value for the Thevenin equivalent resistance, R_{TH} (in $k\Omega$), for the A-B terminal.

5.64 ✔

CHECK SAVE *You have used 1 of 4 submissions*

FINAL EXAM QUESTION 1



$$V - i_2R_2 - i_2R_4 = 0 \Rightarrow i_2 = \frac{V}{R_2+R_4} = \frac{5V}{12K+4.7K} = 0.2994mA$$

$$V + i_1R_1 - i_3R_3 = 0 \qquad i_3 = I - i_1$$

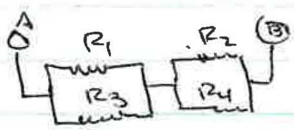
$$V + i_1R_1 - (I - i_1)R_3 = 0$$

$$V + i_1(R_1+R_3) - IR_3 = 0$$

$$i_1 = \frac{IR_3 - V}{R_1+R_3} = \frac{3mA \times 3.3K\Omega - 5V}{7.2K\Omega + 3.3K\Omega} = \frac{4.9V}{10.5K\Omega} = .4667 mA$$

$$V_{TH} = V_{AB} = i_1R_1 + i_2R_2 = .4667mA \times 7.2K\Omega + .2994mA \times 12K\Omega = 6.953V$$

R_{TH}



$$= \left[\frac{1}{7.2} + \frac{1}{3.3}\right]^{-1} + \left[\frac{1}{12} + \frac{1}{4.7}\right]^{-1} = \boxed{5.64K\Omega}$$

$$I_N = \frac{V_{TH}}{R_{TH}} = \frac{6.953V}{5.64\times10^3\Omega} = \boxed{1.23 mA}$$

Q2

We are given a black box that contains only linear circuit elements and a pair of ports. We conduct the following two experiments with this black box.

1. With the right port open, we applied $V_1 = 2V$ to the left port and measured $I_1 = 4mA$ and $V_2 = 8V$. See Figure 2-1.

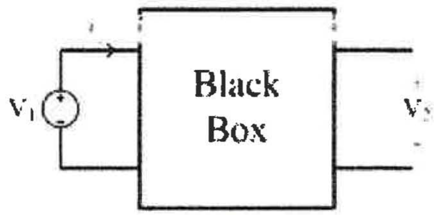


Figure 2-1

2. With the left port shorted, we applied $I_2 = -10mA$ to the right port and measured $I_1 = 5mA$ and $V_2 = -10V$. See Figure 2-2.

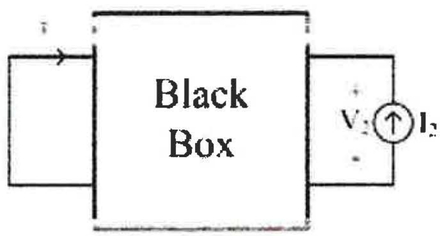


Figure 2-2

We connect a current source to the right port and a voltage source to the left port. This setup is shown in Figure 2-3.

We measure $I_1 = 8mA$ and $V_2 = 4V$. Calculate the numerical values of V_1 and I_2.

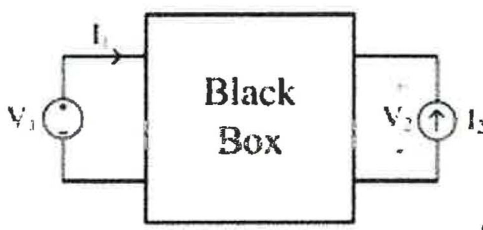


Figure 2-3

	Input		Output	
	V_1	I_2	I_1	V_2
case 1	2V	0	4mA	8V
case 2	0	-10mA	5mA	-10V
case 3			8mA	4V

Linear superposition of outputs

		output
a · case 1	=	a · 4ma + a × 8V
b · case 2	=	b · 5ma + b × (-10v)

$\Rightarrow 4a + 5b = 8$

$8a - 10b = 4$

$\div 2$: $4a - 5b = 2$

$\ominus$ $10b = 6$ $\quad b = .6 \rightarrow a = \frac{5}{4}$

$V_1 = 2.5V$, $I_2 = -6mA$

(a) V_1 (in V) =

2.5 ✓

(b) I_2 (in mA) =

-6 ✓

Q3

You are given a $6V$ battery that is assumed to be an ideal voltage source and a semiconductor diode whose i-V characteristic is shown in Figure 3-1.

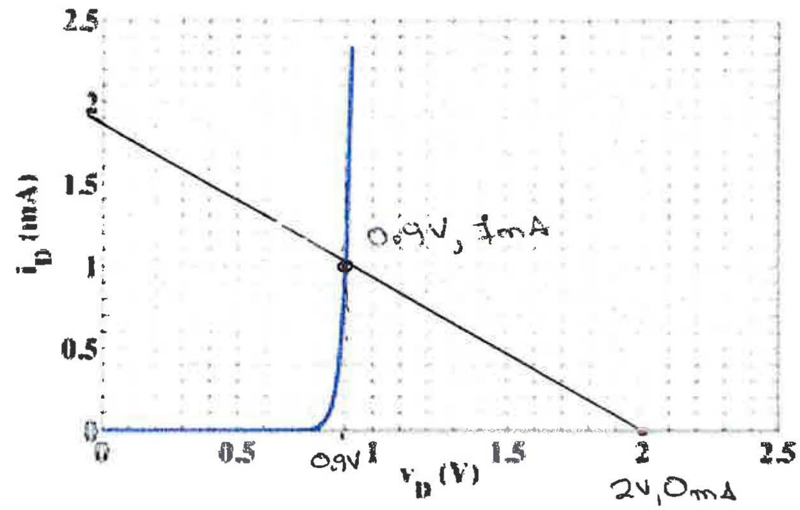


Figure 3-1

You are to design a network of resistors to be used in the circuit shown in Figure 3-2, such that $i_D = 1mA$ when the diode is connected, and $v_S = 2V$ when the diode is disconnected.

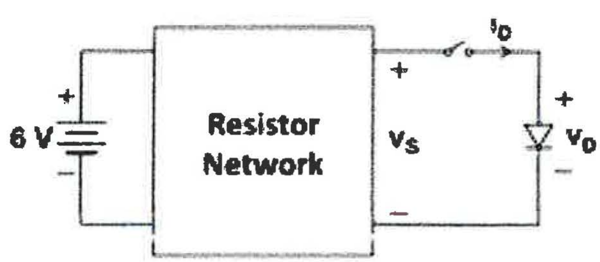


Figure 3-2

(a) If you were to draw a load line that satisfies the above design constraints, what would be its slope in mA/V? Calculate its absolute value (i.e. omit the sign).

slope = $\frac{1mA - 0}{2V - 0.9V} = \frac{1}{1.1}$ mA/V

Consider the Thevenin equivalent circuit of the battery and resistor network that would provide the load line in part (a), shown in Figure 3-3.

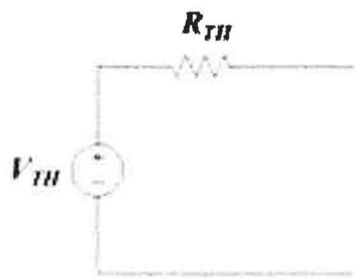


Figure 3-3

(b) Calculate the numerical value of V_{TH}, in V.

2 ✔ = VOLTAGE WITH NO LOAD

(c) Calculate the numerical value of R_{TH}, in $k\Omega$.

1.1 ✔ [illegible] LOAD LINE

Now consider the two-resistor network shown in Figure 3-4.

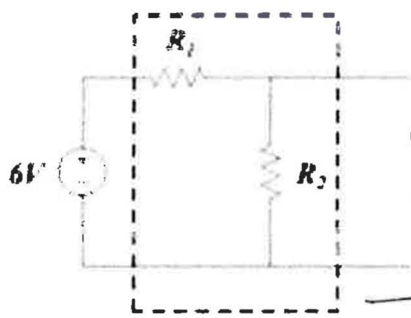
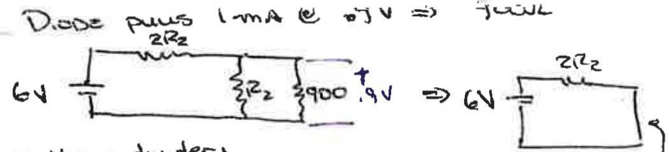


Figure 3-4

VOLTAGE DIVIDER, NO LOAD

$$\frac{R_2}{R_1+R_2} = \frac{2V}{6V} = \frac{1}{3} \Rightarrow R_2 = 2R_1$$

Diode pulls 1 mA @ .9 V ⇒ [illegible]



Voltage divider:

$$\frac{\frac{900 R_2}{900+R_2}}{\frac{900 \cdot R_2}{900+R_2}+2R_2} = \frac{.9}{6} = 0.15$$

$$\frac{900 \cdot R_2}{900+R_2}$$

$$\frac{\frac{900 \cdot R_2}{900+R_2}}{\frac{900 R_2 \; 2(900+R_2)}{900+R_2}} = 0.15$$

$$900 = 0.15\left(900+1800+2R_2\right) = 405 + .3R_2$$

$$495 = .3R_2$$

$R_2 = 1650\,\Omega$ 1.65 kΩ

$$R_1 = 2R_2 = 3300\,\Omega = 3.3\,k\Omega$$

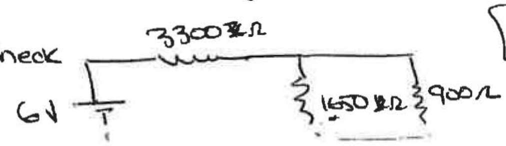
(d) Calculate the numerical value of R_1, in $k\Omega$.

3.30 ✔

(e) Calculate the numerical value of R_2, in $k\Omega$.

1.65 ✔

check



$$\left[\frac{1}{900}+\frac{1}{1650}\right]^{-1} = 582$$

$$\frac{582}{3300+582} = 0.15$$

Q4

Consider the linear circuit shown in Figure 4-1.

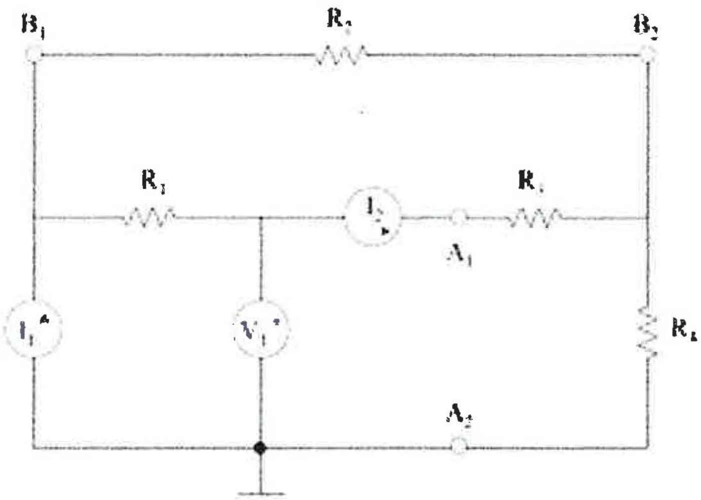


Figure 4-1

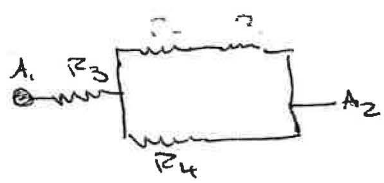
The elements in this circuit have the following values: $R_1 = 7.2k\Omega$, $R_2 = 12k\Omega$, $R_3 = 3.3k\Omega$, $R_4 = 4.7k\Omega$, $V_1 = 5V$, $I_1 = 3mA$, and $I_2 = 7mA$.

Calculate the numerical value of the Thevenin equivalent resistance, R_{THA} (in $k\Omega$), at the A_1A_2 terminal pair.

7.08 ✔

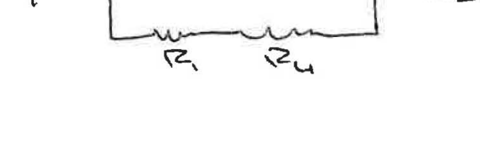
Calculate the numerical value of the Thevenin equivalent resistance, R_{THB} (in $k\Omega$), at the B_1B_2 terminal pair.

5.97 ✔



$$R = R_3 + \left(\frac{1}{R_4} + \frac{1}{R_1+R_2}\right)^{-1}$$

$$= 3.3 + \left(\frac{1}{4.7} + \frac{1}{7.2+12}\right)^{-1} = 7.076k\Omega$$



$$R = \left[\frac{1}{R_2} + \frac{1}{R_1+R_4}\right]^{-1} = \left[\frac{1}{12} + \frac{1}{7.2+4.7}\right]^{-1} = 5.97k\Omega$$

Q5 (15/15 points)

A linear circuit is shown in Figure 5-1.

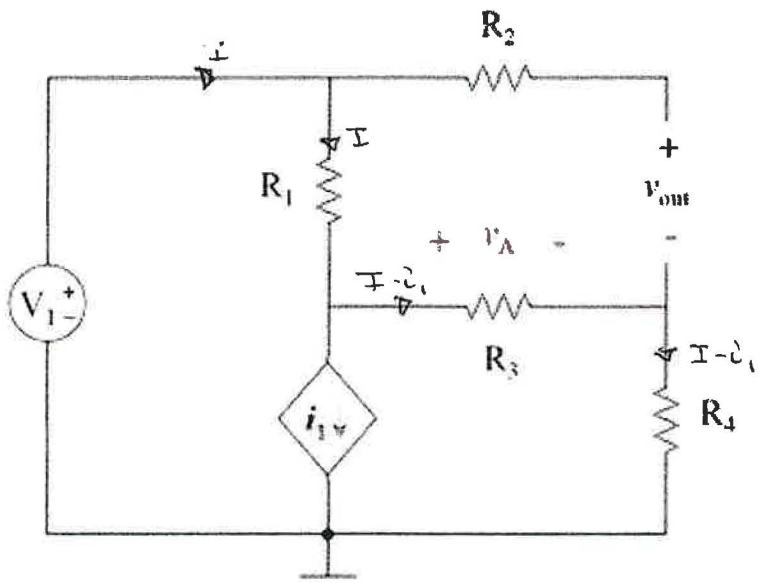


Figure 5-1

The VCCS (voltage controlled current source) is defined as $i_1 = \beta \cdot v_A$.

The other elements in this circuit have the following values: $R_1 = 3.3k\Omega$, $R_2 = 8k\Omega$, $R_3 = 1.8k\Omega$, $R_4 = 1.8k\Omega$, $\beta = 1.5mA/V$, and $V_1 = 5V$.

(a) Calculate the numerical value of the voltage v_A in V.

0.569 ✔

(b) Calculate the numerical value of the voltage v_{OUT} in V.

4.43 ✔

(c) Calculate the numerical value of the Thevenin equivalent resistance, R_{TH} in $k\Omega$, as seen from the terminals of v_{out}.

9.16 ✔

ANSWER TO FINAL EXAM QUESTION 5

KVL: $+V - IR_1 - (I-i_1)R_3 - (I-i_1)R_4 = 0$

$$V - IR_1 - (I-i_1)(R_3+R_4) = 0$$

$$i_1 = \beta \cdot V_{R3} = \beta\,(I-i_1)R_3 = \beta R_3 I - \beta R_3 i$$
$$i_1(1+\beta R_3) = \beta R_3 I$$

$$i = \frac{\beta R_3}{1+\beta R_3}\cdot I \qquad = I - i = \left(1 - \frac{\beta R_3}{1+\beta R_3}\right)I = \left(\frac{1+\beta R_3-\beta R_3}{1+\beta R_3}\right)I$$

$$I - i = \frac{1}{1+\beta R_3}\,I$$

~~For ... ~~

BACK TO KVL: $V - IR_1 - \frac{R_3+R_4}{1+\beta R_3}\,I = 0$

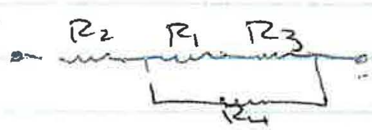
$$V = I\cdot\frac{(1+\beta R_3)R_1 + R_3 + R_4}{1+\beta R_3}$$

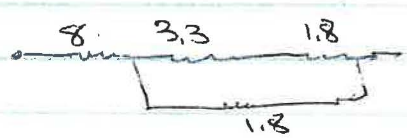
$$\Rightarrow I = \frac{1+\beta R_3}{(1+\beta R_3)R_1 + R_3 + R_4}\cdot V = \frac{\overbrace{1+(1.5\times10^{-3}\,\Omega^{-1})(1.8\times10^{3}\,\Omega)}^{3.7}}{3.7\times 3.3\,k\Omega + 1.8\,k\Omega + 1.8\,k\Omega}\cdot 5V$$

$$= 1.17\ mA \qquad \Rightarrow (I-i) = \frac{1}{1+\beta R_3}\cdot I = \frac{1}{3.7}(1.17\,mA) = .316\,mA$$

$$V_A = (I-i)\cdot R_3 = 0.316\,mA\times 1.8\,k\Omega = \boxed{0.569\ V}$$

$$V_{OUT} = V_{R_1} + V_A = IR_1 + V_A = 1.17\,mA\times 3.3\,k\Omega + .569\,V = \boxed{4.43\,V}$$

R_{TH} seen by V_{OUT} =


=


$$= 8 + \left(\frac{1}{5.1} + \frac{1}{1.8}\right)^{-1} = 9.16\ k\Omega$$

Q6 [illegible]

The logic function $F = A + C - B \cdot D + C \cdot D - A \cdot C \cdot D$ can be represented by the truth table below in Table 1.

A	B	C	D	F
0	0	0	0	d_0
0	0	0	1	d_1
0	0	1	0	d_2
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	d_3
1	0	1	0	
1	0	1	1	d_4
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Table 1

The truth table has five outputs labeled as d_i with $i = 0, 1, \ldots, 4$.

(a) Enter the boolean value for d_0.

1 ✔

(b) Enter the boolean value for d_1.

0 ✔

(c) Enter the boolean value for d_2.

1 ✔

(d) Enter the boolean value for d_3.

1 ✔

(e) Enter the boolean value for d_4.

0 ✔

Q7 (15/15 points)

Consider the diode D_1 shown in Figure 7-1.

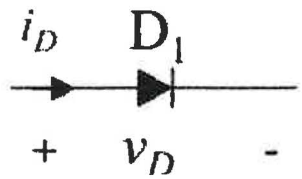


Figure 7-1

The diode is a non-linear device with the following I-V characteristics:

$$i_D = I_S \cdot \left(e^{\frac{v_D}{V_{TH}}} - 1\right)$$

where i_D is the current through the diode, I_S is the reverse bias saturation current, v_D is the voltage across the diode, and V_{TH} is the thermal voltage. Assume $I_S = 8 \cdot 10^{-12} A$ and $V_{TH} = 20mV$.

Although the diode is a non-linear device, it can be modeled as a resistor for small signals. Calculate the numerical value of its small signal resistance, in Ω, when the DC bias current is $I_D = 10mA$.

2 ✔

$$i_D = I_S\left(e^{\frac{v_D}{V_{TH}}} - 1\right) \Rightarrow \frac{i_D}{I_S} + 1 = e^{\frac{v_D}{V_{TH}}}$$

$$\Rightarrow V_D = V_{TH} \ln\left(\frac{i_D}{I_S} + 1\right) = 0.020 \ln\left(\frac{10^{-2}}{8\times10^{-12}} + 1\right) = 0.41893V$$

$$\frac{di_D}{dv_D} = \frac{I_S}{V_{TH}} e^{v_D/V_{TH}}$$

$$\frac{dv_D}{di_D} = \frac{V_{TH}}{I_S} e^{-v_D/V_{TH}}$$

$$R = \left.\frac{dv_D}{di_D}\right|_{\substack{i_D = 10mA \\ V_D = 0.41893V}} = \frac{.020}{8\times10^{-12}} e^{-\frac{.41893}{.02}} = 2$$

Behavioral Economics in Action

University_of_TorontoX: BE101x Behavioural Economics in Action

This course was interesting to me because part of its focus is how to get people to do what is in their own best interest. (On the flipside, one could use these same tools to get people to do what is *not* in their own best interest. It used example throughout the course about getting people to save for retirement, lose weight, make better decisions about healthcare and the like.

Each of the above is similar to the challenge I face with my students. I know that they will do better in the course if they do their homework; study and do problems over the course of each week rather than waiting until the last minute; come to class having already read the preparatory material and having done the sample problems in the textbook; and write up their lab blogs and submit them close to when the actually finish making measurements in the lab.

And the crazy thing is that *my students*! Yet all of the evidence (I can see online when they submit their work, and they readily discuss when they get things finished) indicates that more than 90% of them wait until the last possible moment to submit their work and as a result have no "elasticity" in being able to ask for help when they run into problems.

My starting assumption is that almost none of my students comes into my Physics course (or any course, really) class with the goal of getting a C. But their behavior right from the outset tend to undermine their goals.

What motivated me to take this course was a desire to put together the default reward structure in my Physics course so that students would make the "right" choices.

- How do I make sure that they watch the videos that I am preparing and work the problems that I want them to work before they get to class?
- How do I make sure that they get their lab blogs/writeups done in a timely manner?
- How do I make sure that they are doing their homework on time?
- How do I make it easier for them to prepare for exams?

There were six "lessons" or "weeks to the course. I completed five of them, then went out to buy the book that the course was based on: ***Nudge: Improving Decisions about Health, Wealth, and Happiness***, by Richard Thaler and Cass Sunstein. After that I had no real need to complete the course—I had gotten from the course what I wanted.

The biggest lesson I took home from this course was that I had to overtly and explicitly incentivize the behaviors I wanted (so, make 20% of the course grade related to watching the videos and coming in having done the problems embedded in them) rather than hoping that students would be "mature" enough to do the "right thing." And that if there was something required for students to have to be successful, make getting that thing as simple as possible. (For example, all students are required to have a notebook for their "embedded video lecture" problems and another for their online homework solutions. I went to Staples and got 150 notebooks (17¢ each) and just gave them to student the first day. Miraculously, every had two notebooks for the second day of class!

Lesson 1

This course is essentially about how to get people to change their behavior. It lists four basic approaches:
1) Ban something
2) Use a carrot and stick approach—positive and negative incentives
3) Advertising and Knowledge—attempt to change behavior by reframing a behavior
4) Apply a "nudge"—this is an approach that considers the person constructing an environment as a kind of "choice engineer". The idea is to make it easier for someone to make the "right" choice, while not actually limiting their ability to make a different choice.
--One way is to make the desired choice the default choice—if you do nothing, you are automatically signed up as an organ donor. This is called a nudge. defined as:
"A nudge is any aspect of the choice architecture that alters people's behavior in a predictable way without forbidding any options or significantly affecting their economic consequences. To count as a mere nudge, an intervention must be easy and cheap to avoid. Nudges are not mandates." So a nudge is a deliberate change in the choice architecture with the goal of engineering a particular outcome.

There was a description of "normal" economics—that is, the assumptions that underly standard economics theory about what goes into rational decision making:
1) Complete information about each choice, the cost of each choice, and the probability of each outcome
2) Cognition, rather than emotion drives choices—neither guilt, nor regret, nor sympathy guide decision making
3) an infinite Computational ability to determine the best choice given knowledge of all of the information and possibilities
4) Consistency with the axioms of choice, which means that if choice A is preferable to B, and B is preferable to C, then A is preferable to C, and if A and C are rationally equivalent, there will be no preference for choice A over choice B

The course then goes on to explain how people do NOT necessarily make the "rational" choice. The overall theme of the course is that each of us is really occupied by two minds—one rational and self controlled, and the other driven by habit, appetites, and mindlessness kind of like the devil on one shoulder and the angel on the other.

Once example was that, given three choices of cup size for coffee, people usually choose the middle size, explaining that the small cup isn't enough but the large cup is too much. When the smaller cup is removed and an even larger cup is made available, people will *still* choose the middle cup size and give the same reason, even though now the middle cup size is the size that use to be "too much" coffee. This was called the CONTEXT EFFECT.

Another example was the idea of decision points—if you go to a movie and are given a large bucket of popcorn, most people will eat it all= one decision point, right at the beginning.
If you split the popcorn into four bags, you have four decisions to make. People still finish a bag, but then they have to decide if they really want more. If they have to actually open the bag to access more popcorn they are even less likely to eat more. So one way to design a nudge to discourage an undesirable behavior is to create multiple decision points. This is essentially the idea behind putting the chocolate in a locked cupboard, where you have to get a key in order to

eat some, or by making it difficult for people whose politics differ from your own to be able to vote. You don't actually *forbid* it—you just make it difficult enough that statistically, fewer people will tough through the required steps.

Lesson 2 discussed several more ideas:
--mental accounting—how people artificially segregate things that are really identical (putting money into a separate account for vacations, even though it is just a click away)
--sunk cost effect—how people are loathe to not do something or consume something that they have already paid for.
--choice overload—people do not have infinite cognition and infinite calculation capacity, so if you give them too many choices they just shut down. In a choice architects desire to provide as much information as possible the effect is to discourage people from making any choice (which is why the proper choice of a default is so important)
Examples here are
--having a website that asks you a series of questions and calculate a small set of recommendations ranked based on the information you provide (choosing a health plan, for example.)
--dropping unpopular or seldom chosen options from a list
--having folks choose one from column A, one from column B, etc.
The example was given of a Chinese restaurant that had 85 items on the menu but people usually chose the first 4 or five. The solution was to break the decision into
++what kind of meat/protein do you want
++what kind of sauce
++what kind of vegetables
++what kind of noodles
++what level of spiciness

Discounting
—the idea that people value things differently now than they will in the future
--the idea that people feel losses much more than they feel gains

Arousal—the smell of cookies can make people behave impulsively
Self-control strategies:
--get people to precommit for a future savings behavior, then figure out how to "lock" it in
--Social contract—if people publicly proclaim a goal, you can set up a "judge" as to how well they are doing, a "cheering section" and an accountability report that can, say, post to Facebook that you failed
--set up a savings account where you have to get a key from someone else in order to access the money you put away
--strategies that make it hard to overconsume
--freezing your credit card into a block of ice
--only taking three cigarettes with you to work

Lessons 3&4:

building blocks of what makes a scientific experiment on behavior, ethics etc.
(Not particularly interesting, in that I am pretty solid on scientific method already.)

Lesson 5:
Strategies of behavior change, revisited:
1) Impose restrictions and restraints (government mandates, for example)
2) Use a carrot and stick approach—positive and negative incentives (credits or taxes)
3) Marketing and persuasion—attempt to change behavior by reframing a behavior
4) Apply a "nudge"—make it easier for people to accomplish the desired behavior

Examples:
--Charge people 5¢ for a plastic bag (can be considered a tax)—not clear if the fact that the actual cost matters, just that there is any cost at all
--people in India were more likely to follow economic advice if they had to pay for it, no matter how small the price was
--raising the cost of energy doesn't affect consumption much—the payment doesn't happen at the same time as the consumption does
--if you want people to actually get their annual physical checkup, you can make it free (an economic strategy) but give them a ticket for a free checkup that expires if they don't use it (fear of loss)

Nudge design:
You can give people better information to guide their decisions, and you can design a choice architecture that steers them toward the desired decision
--requires a clear understanding of the mistakes people make and what you can do to help them avoid those mistakes
--requires making sure that access to tools and information and the desired outcome is clear and unobstructed
--requires running experiments to see what works

Nudger's tool kit
--Active choice, enhance active choice
  --active choice—are you going to get a flu shot
  --enhanced—Yes I will protect me and my family, or No, I will expose myself and my family to the risk

--Anchoring—people who encounter high prices early in their shopping experience are more likely to by the less expensive item later. You can artificially set expectations by exposing them to a more expensive choice first

--Asymmetric dominance Decoy
If A is better than B on one attribute, and B is better than A on another attribute, you can offer a decoy choice C which is worse than B on both attributes. This shifts the choice to B

Headphones example:

Headphones	Sound	comfort
A	100	50
B	50	100
C (decoy)	40	90

Adding C to the mix increases shifts the preference to B
--Automatic enrollment (but with and option to withdraw later)

--Choosing vs. Rejecting:
If you are choosing between two job applicants:
A, who is average in 4 categories
B , who is superior in 2 and weaker in 2
If you are asked to choose one, folks will choose B (emphasizes reasons to choose)
If you are asked to reject one, folks will choose A (emphasizes reasons to reject)
--Compromise effect—folks tend to choose the middle option
--Construal levels—if someone is planning, they are focusing on future benefit. If they are executing, they are focusing on immediate inconvenience
--Decision points
--Opt in vs. Opt out (default values)
--Earmarking—physically segregating money or some other resource
--Framing (gain vs. loss)—a cash discount is more acceptable than a credit surcharge.
--Goal visibility—keep reminding folks constantly of what the goal is—it has to be in sight
--pain of payment and transparency of cost—credit is easier to spend then cash

Payment depreciation—the pain of not getting what you paid for decreases with time

Peer programs/social comparison—if everyone else is doing something, folks are more likely to do it too.

Perceived progress—people are more motivated if they can see their incremental improvement or gain

Self-awareness—identity—require people to sign "everything in my tax return is true" *before* they start working on it.

There was more discussion of various ways to design nudges.
I decided at this point to buy the book that the course was based on: ***Nudge: Improving Decisions about Health, Wealth, and Happiness***, by Richard Thaler and Cass Sunstein.

At this point I decided that I didn't really need to finish the course. I could watch the last section of videos but I figured I had what I needed.

Other Learning Activities

OTHER LEARNING ACTIVITIES

SPOT course at Mt. SAC. Got my Distance Learning Certification
Did the Moodlerooms gradebook and discussion modules (3-5)
Met With Michelle Newhart to learn how to accomplish a video, reading and quiz in Moodlerooms. She suggested that I use the "Lesson" format in Moodlerooms.

Wrote out a quick version of an introduction to Vectors, with various quick quiz/check-in bits, and tried entering these into Moodlerooms. It is really tedious to enter a multiple-choice question or a matching question, especially if one choice can be used more than once.

Did the CS5 course—July 5th – September 21st
Started on the VPython site—went through the set of 8 or so tutorials on there on how to draw various shapes, arrows, then tried animating things

Python Numpy graphing and data manipulation package:

Numpy is a package of programs available in Python that allow you to generate graphs and manipulate potentially vast arrays of data. I wanted to know how to generate graphs in real time to match stuff I was illustrating in V-python. The Numpy library is available from a software company named Enthought. So I downloaded their incarnation of Python called Canopy.

They have 46 online lectures on using Numpy, and another 20 lectures on SciPy, which allows you to do all sorts of very high-level mathematical manipulations and produce very cool graphs.

I spend a week and a half or so going through about half of the lectures, writing the programs and trying things out. I figured out eventually that this was a much higher level of detail than I wanted to get into. I was able to figure out how to make the graphs I wanted to make. Everything else was more than I needed.

Adobe Illustrator

I met with Brian Bouskill in the Graphic Design dept for a couple of hours to see about how to do animations in Adobe Illustrator and After Effects. He showed me a lot of cool things. I spent about a week and a half on Lynda.com going through the Illustrator tutorials on how to use the various tools. I got about 85% of the way through and realized, again, that this was a very powerful drawing program and that I would need to dedicate some serious time to learning how to use it if I were going to do stuff more complex than what I could already draw in Microsoft Word. I did end up making a couple of nice drawings in Illustrator, but most of what I used in my lectures was either free images downloaded from the web, things I drew in Microsoft Word or PowerPoint, or sketches I made by hand.

Other online courses through Lynda.com
Introduction to Screencasting
iMovie 10.0.2 training

Programming an Arduino

An Arduino is a microprocessor that can be programmed to control and take data from a variety of sensors. I went through the course on the adafruit website. This course turned out to be very simple, and I was able to get a variety of LED's to blink in a desired fashion.

The other course I started was more rigorous. Each lesson involved getting some more electron parts (memories, display cards). The last lesson I did required first soldering 30 or 40 small wires to a corresponding pin on an LED display card. I did that, but I am not good at soldering, and the display didn't work. At this point I realized that it could be either my soldering or the display itself that was at fault, and this kind of fault diagnosis is not my idea of interesting. I abandoned the course at that point.

Solving problems at a level a step beyond what we teach

Online Electrical Engineering course—please refer to the material in a previous section of this "tab" on Other Learning Activities. I chose here to take an online course rather than simply solve problems from a textbook. This material will be useful to me in better understanding what our students do in our Engineering 44 (Electrical Engineering) course, taken by some of our students after having completed our Physics 4B course.

Working problems in Special Relativity

Edwin Taylor wrote in the sixties a book on Special Relativity called *Spacetime Physics.* The way we usually teach special relativity at our level is to consider the Lorentz transformation, which is a set of equations for comparing measurement made in one frame of reference (say, seeing yourself at rest in the lab) with measurements made in another frame of reference (Say, a spaceship going by at a large, constant speed.)

Taylor's approach is to look at an invariant quantity called the *interval.* This quantity is the same in all inertial (constant velocity) frames of reference. His "take" is that time is another dimension, different but similar to position, and that our historical tendency of looking at space and time as totally divorced from one another makes it hard to understand special relativity. He argues convincingly that both times and distances should be measured in units of seconds. The book has three chapters to it. I downloaded the first chapter from his website (www.eftaylor.com/special.html) and worked through all of the example problems within the chapter and most of the problems at the end of that chapter.

This approach to Special Relativity that is definitely one level up from what we do in our Physics 4C course. Its approach to "time dilation" and "length contraction" is more sophisticated and generally applicable than what we teach.