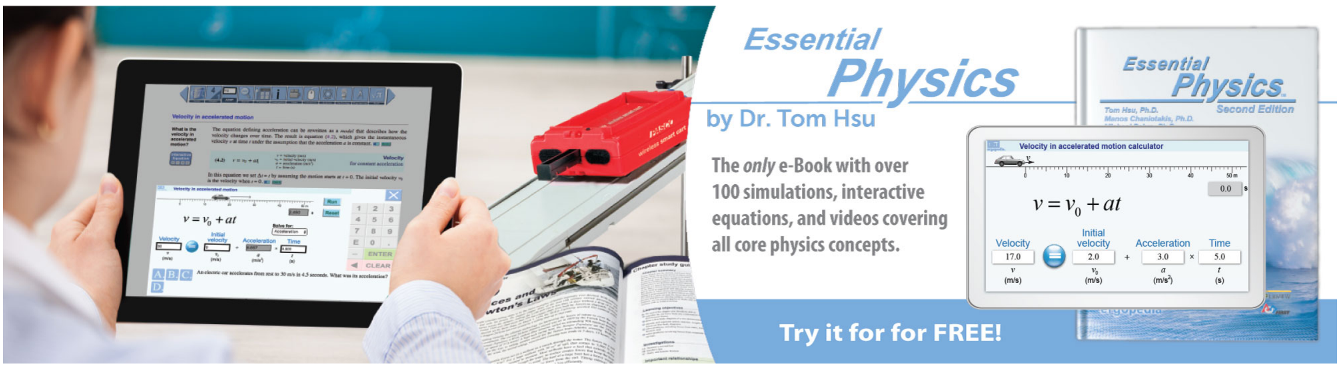
Applications of Macintosh microcomputers in introductory physics

Robert J. Beichner

Citation: The Physics Teacher **27**, 348 (1989); doi: 10.1119/1.2342789
View online: http://dx.doi.org/10.1119/1.2342789
View Table of Contents: http://aapt.scitation.org/toc/pte/27/5
Published by the American Association of Physics Teachers



Applications of Macintosh Microcomputers in Introductory Physics

By Robert J. Beichner

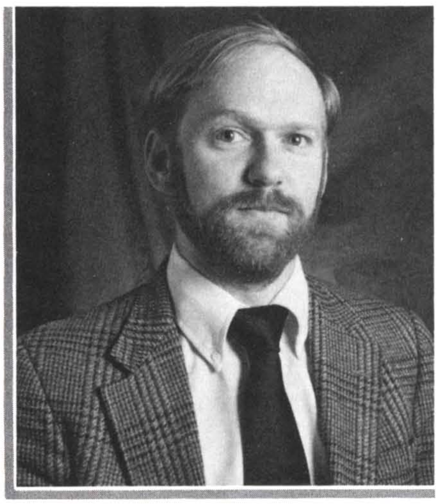
y intention in this article is to highlight some of the exciting projects now being undertaken with state-of-the-art microcomputers in physics labs. For the most part, I will be referring specifically to Macintosh computers, although the Amiga and newer IBM microcomputers have similar capabilities. Aside from the greater memory and speed, what sets the Macintosh apart from earlier micros is its user interface and graphics capabilities. Software operation is usually icon based with pull-down menus, multiple windows, and control via a mouse. Probably most important of all the user interface features is the consistency found from one application (program) to the next. This is particularly important in education. Normally, we want the student to use the computer as a tool. Long, involved instructions on how to use a particular piece of software detract from the focus on the instructional topic. With a consistent interface to each program, students quickly learn how to use a wide variety of software and gain an intuitive approach to solving problems with the computer.

Availability

These computers are making substantial headway into education, particularly at the university level. Their entry into this market is not only enhanced by their features but also by manufacturer incentives. Schools that are members of Apple Computer's University Consortium often have hundreds of Macintosh computers. Several universities report more than 10,000! Their use in pre-college instruction has been limited so far, probably because of higher equipment costs and the perception that color is necessary to educational computing. Also, elementary and secondary level institutions may not feel they have a need for the increased computing power these machines offer.

Software Development

Within the past few months a greater variety of Macintosh CAI packages have become available. There are some interesting reasons for this sudden upswing in educational software production. Until recently it was very difficult to develop applications for Macintosh-class machines. This is partly because so much is expected by the software users. If the operation of a program is not intuitively obvious, it simply is not used. This is certainly a different state of affairs compared to the more standard microcomputer offerings. With them, each program interacts with the user in its own (i.e., unique) way. Learning to operate two programs is very nearly twice as much effort as learning to use one. This is not the case with the consistent interface of the Macintosh. However, this ease of operation exacts a cost from developers, "because you have to deal so intimately with so many details that are vital to the operation of the User Interface Toolbox, it's very difficult to completely divorce yourself from what's happening at the low level."[1] In other words, Macintosh computers may be easy to use, but they have been very hard to program. This situation seems to be changing. Newer programming languages and environments are using the power of the computer to make life simpler for software developers. Because of this, we can expect to see a rapidly increasing number of educational programs.[2] This is significant since it has



Robert J. Beichner *received his B.S. degrees in physics and mathematics from Pennsylvania State University and his M.S. in physics from the University of Illinois–Urbana. He is nearing completion of a Ph.D. in science education at the State University of New York at Buffalo. His research interests include the impact of technology on science instruction. He is the associate director of SUNY/Buffalo's Center for Learning and Technology and teaches physics at Erie Community College in Buffalo.*

been acknowledged by experts[3] that CAI software for the Macintosh is superior to that available for other computers. I think it is safe to say that we will be witnessing a "blooming" of high-quality, computer-assisted instructional materials in the near future.

People writing much of this new software seem to prefer C and Pascal as development languages. Fast compiling time helps speed up the authoring process. Other languages in use include BASIC and Forth. There is also a specialized "real-time Pascal" (Rascal) designed for running equipment interfaces on the Macintosh.[4] The language used depends on the developer's requirements. C has the advantage of higher running speed and compact compiled code. Pascal doesn't require as thorough an understanding of the workings of the computer. "Pascal helps you more by not allowing weird things. C is a bit more of the professional's programming language."[5] BASIC is very easy to start using, but does have somewhat limited power. Apple has come out with a generic application "shell," which promises to make it substantially easier to incorporate the Macintosh interface into user-written software for their computers.[6] The complicated handling of windows, menus, mouse control, etc. is somewhat automated. The net effect of all these newer programming environments should be increased productivity from programmers and the addition of more teachers to the ranks of educational software authors.

Perhaps an even more promising development is HyperCard®. Although somewhat difficult to describe, this new system software for the Macintosh brings programming to a new level of simplicity. The person using HyperCard to build a lesson uses computerized tools shaped like brushes and pencils to draw pictures of what is being discussed. Alternately, a scanner or digitizer can be used to capture images. Then "buttons" that link this picture (or specified parts of it) with other related information can be quickly and easily added. For example, if a laboratory spectrometer was being presented, clicking on the grating might result in the computer giving the student the slit separation or an explanation of how a grating works. Clicking on the eyepiece might display a simulated series of spectral lines. Using the mouse to select one of the lines could then display the wavelength of that line and perhaps even a drawing showing the electron energy levels involved. With HyperCard it becomes almost trivial to set up a "free-form" database, which allows the student to investigate items of interest in whatever order they choose. (Experience at SUNY/Buffalo has indicated that CAI projects that have been in development for months can sometimes be redone in a matter of *days* using HyperCard, with better results than the original BASIC or Pascal implementation! Additional commands and functions can also be added to HyperTalk, the programming language that runs HyperCard). HyperCard programs, called "stacks," can run on any Macintosh with at least one Megabyte of RAM. I think you will see some very exciting things for the physics classroom being done with HyperCard. For the teacher new to the Macintosh, HyperCard is definitely the best way to start.

Applications

These advanced computers have been used for a wide variety of applications in higher education. These range from simple word processing tasks (done by just about every school owning these machines) to operating a Cray supercomputer through a Macintosh interface.[7] The rest of this article will discuss some of these applications. The data upon which this material is based came from personal communications with faculty members, computer support staff, and computer and interface manufacturers. A great deal of information was also garnered from the pages of *Wheels for the Mind*, a publication prepared at Boston College for Apple Macintosh users in higher education.[8] It is published quarterly and contains very helpful summaries of projects being done at various academic institutions. If you would like to keep up to date on how this particular computer is being used in education, it is an indispensable resource.

Another good place to look for Macintosh CAI materials is in Kinko's Academic Courseware Exchange catalog.[9] Here you will find many excellent programs at very reasonable prices. The whole premise of this catalog is to offer high-quality educational software at textbook prices. It is my impression that the software available for these computers is often lower in price than similar programs for other types of micros. Keep in mind that Macintosh software is usually better and easier to use than "equivalent" programs for other machines. This results in real value and may help offset the higher equipment costs.

Simulations

One of the physics packages offered in the Kinko catalog is *Mechanics* by Blas Cabrera of Stanford University (see Fig. 1). This package includes a variety of programs allowing students to investigate projectile motion, relativity, harmonic oscillators, Kepler's Laws, and potential wells. For example, using the Kepler program, students can collect data on planetary motion, something which might be very difficult to do in the "real world." Another interesting part of this package is the ballistics program. Students can use it to investigate the effects of wind resistance and see how the launch angle for maximum projectile range is affected by it. In short, these simulations allow students to gather data on situations that would be difficult to actually produce in the typical student lab. Beyond this, they permit easy adjustment of parameters in order to more easily discover their effects. Learning to control variables so that only one varies at any given time is an important lesson that is often missing from conventional student lab experiences.

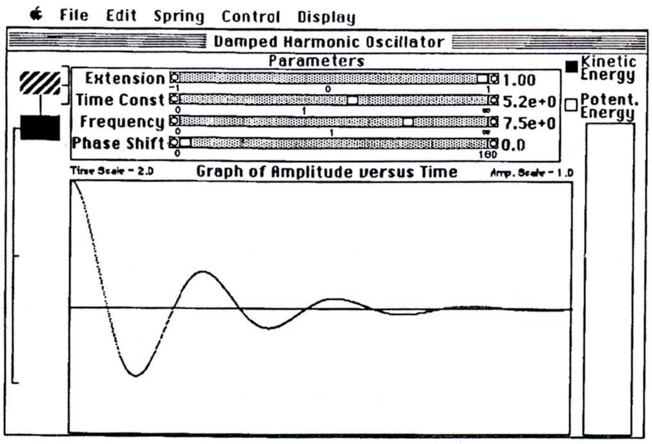


Fig. 1. Sample screen from *Mechanics* by Blas Cabrera. The four horizontal bars along the top allow students to adjust parameters with a simple movement of the mouse.

The idea of simulating hard to reproduce situations is perhaps best exemplified by the work of Eric Berman and Edwin Taylor of the Massachusetts Institute of Technology. They have produced a simulations package called *Spacetime Software.*[10] By using this set of programs, students are able to study relativistic effects that are really only evident at extremely high speeds (see Fig. 2). Excellent use of animation and easy to change parameters allow students to create their own universe for exploration. Even teachers remark about how this simulation has helped them understand situations that previously could only be visualized in the mind's eye.

Another lab possibility involves the construction of spreadsheet models. This allows the students to actually set up the formulas required by the model. The students may lose some of the fancy graphical output of a specialized simulation program, but they get an in-depth introduction to the ideas of modelling. This sort of work has been carried out by Robert Fuller of the University of Nebraska.[11,12] For example, students take

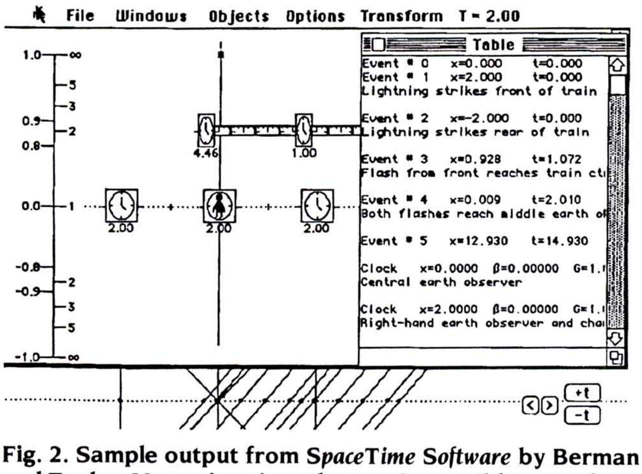


Fig. 2. Sample output from *SpaceTime Software* by Berman and Taylor. Many situations that are impossible to perform in the lab can be simulated, allowing students to get a better feeling for many of the "non-common sense" ideas of relativistic mechanics.

data on the bouncing of a cart on an inclined air track. Using a variety of data processing approaches (i.e., differences between spreadsheet rows, ratios of row values, etc.), the students attempt to find a rule that relates the number of bounces to the bounce height. Doing this sort of thing by hand quickly becomes tiresome. Allowing the computer to help as a "super calculator" makes the analysis much less tedious, and, as a result, the students are more willing to make adjustments to their model.

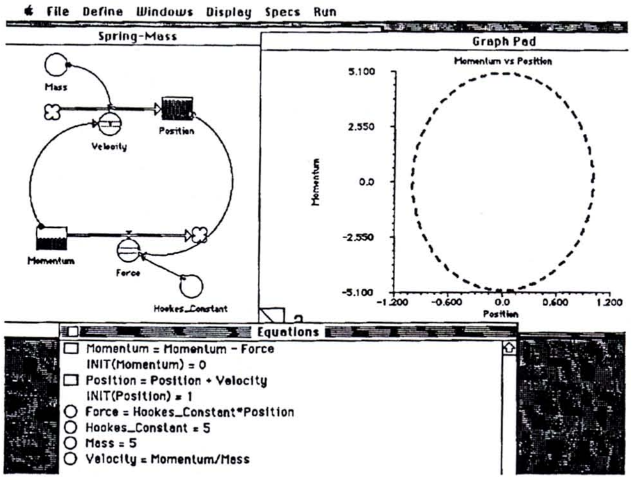


Fig. 3. Sample output from *Stella* from High Performance Systems, Inc. showing a model of an oscillating spring. A variety of graphs can be displayed. This figure shows one of the more interesting ones.

A different type of simulation makes use of a modelling software package called *Stella*.[13] With this program, students can easily control variables related to their own theoretical model of a physical situation. For example, a simple harmonic oscillator can be designed, with or without damping and driving forces, and the students can graphically follow the flow of energy from potential to kinetic and back again (see Fig. 3). This can be done in conjunction with a "real" example of simple harmonic motion to help the student make connections between the more abstract model and the actual physical situation it describes. The impact of this type of lab exercise is discussed by Robert Tinker of the Technical Education Research Center (TERC):

> The best use of models has students create the model rather than simply explore a simulation, a canned model that someone else has created. While simulations based on models others have created have an important role in learning about science, the ability to create a model seems much more central to the goals of science education. Creating a model is a form of theory building, and testing the accuracy of the model goes to the essence of science.[14]

Robert Fuller and David Winch have produced an interesting HyperCard stack called "Guilty or Innocent?"[15] They demonstrated its use at the Conference on Computers in Physics Instruction held at North Carolina State University this past summer. Students using the stack are placed in a situation where they must collect data, interview witnesses, and perform calculations to determine what happened during a simulated car collision. They are then called upon to state their opinions during a mock trial. They must be able to justify their conclusions to the attorney (who actually asks the questions via digitized speech). The stack is fun to use and contains quite a lot of physics.

Data Sampling and Analysis

Many very interesting microcomputer-based labs use these computers to collect and analyze data. Most notable among these is the *Benchtop Instrument*, initially developed at Reed College.[16,17] With this interface, digital or analog signals can be fed into the computer for analysis. Pre-programmed software is available with the equipment, but many people have written their own instrument packages in a real-time language called Rascal. A brief example of the work of Elisha Huggins at Dartmouth demonstrates just how powerful this type of package can be.[18]

Huggins' software works in conjunction with the *Benchtop Instrument* to create a sophisticated, digital-storage oscilloscope. (Thornton Associates[19] is also coming out with a version of what he calls *MacScope*.) Most ordinary oscilloscope controls are available. As Fig. 4 demonstrates, there are other additional features such as a virtually instantaneous Fourier transform capability. Students can also selectively filter out frequencies (with simple mouse commands) and immediately see how this changes the waveform.

Work to develop entire laboratory courses making use of Macintosh computers is underway at several institutions. Drexel University[20] has been a leader in this area. Blas Cabrera of Stanford has developed a set of simulations covering an entire year of college-level introductory physics.[21] The University of Rochester[22] is also developing a data sampling and analysis set for their introductory physics labs. In addition, a variety of

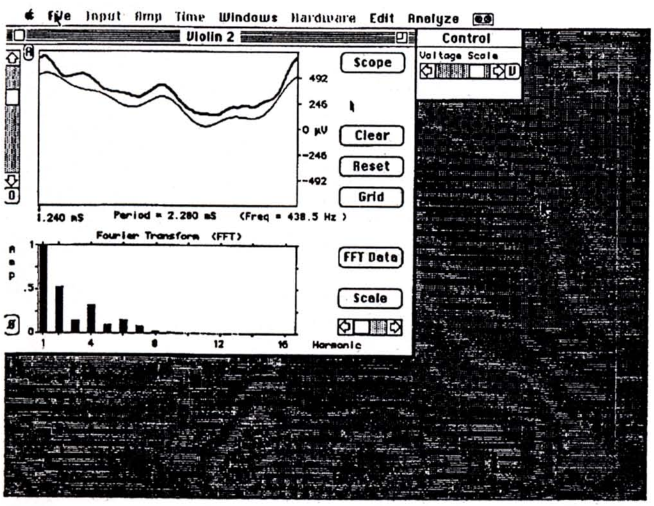


Fig. 4. Screen from *MacScope* by Elisha Huggins. The computer and interface take the place of a powerful oscilloscope with a built in Fourier Spectrometer.

equipment manufacturers are designing interface equipment, which utilize the graphics environment of a Macintosh to allow the "building" of monitoring systems by dragging icons representing instruments such as temperature probes, noise filters, and strip chart recorders around on the screen.

The ability to input video images to the computer adds another dimension to its use in education. Dean Zollman of Kansas State University is using physics videodisks interfaced to an Amiga microcomputer to help students develop models of extended body motion.[23,24] (The Amiga is mentioned since it has somewhat similar capabilities, but not nearly as many applications, as the Macintosh.) This allows them to freeze frames or play them back at any speed, up to real time. The 1/30 of a second time for each frame builds in a time base for speed calculations, etc. Similar work is being done by SUNY Buffalo's Center for Learning and Technology in cooperation with the SUNY College at Buffalo, utilizing a Macintosh II for collecting data from student-produced videotapes. Using *VideoGraph*, a HyperCard stack I have developed, students are able to watch a replay of the videotaped motion on the computer screen while graphs of position and velocity are simultaneously generated (see Fig. 5). For example, students are actually able to watch a projectile reach the peak of its trajectory while the vertical speed graph crosses the time axis. The hope is that this will help the students build the mental links between the physical world and its mathematical representation.

Another application of computer animation has been undertaken by Eric Mazur of Harvard.[25,26] He has used a digitizer to input pictures of proton-proton collisions into a Macintosh. The computer then quickly displays this series of images, making a simulated movie. This allows students to get a "feel" for how these interactions compare to more easily observed everyday collisions.

Bob Tinker and others at TERC are working on developing a general set of probes for temperature, light, force, motion, etc. and adapting them to a variety of microcomputers, including the Macintosh. What has been shown so far is very impressive.[27] It will be interesting to see what the talented people at TERC can produce for a powerful computer like the Macintosh.

Instructional Lessons

Broderbund software has released a nice package called *Sensei Physics*.[28] It contains over 300 different problems from most of the content area of introductory physics. It also briefly discusses the theory and gives examples to help students gain understanding of the basic material. Its clever animations and thought-provoking "exploratory" format make this package a highly motivating one for students; examples are shown in Fig. 6.

Some very noteworthy work has been going on at Carnegie-Mellon University. A team including Bruce Anderson and David Trowbridge are working to adapt

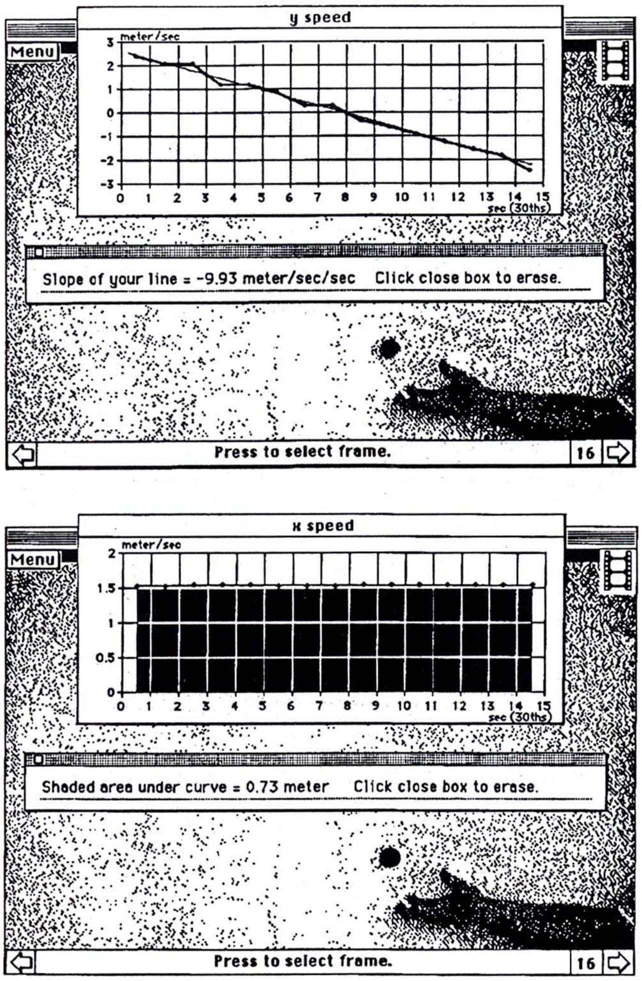


Fig. 5. Screens from *VideoGraph* by Robert Beichner, showing the slope and area calculation features. Not easily shown is the animation of the background image while the motion graphs are produced. This application was written using *HyperCard*, with additional Pascal commands supplementing *HyperTalk*.

cT, an extremely powerful lesson authoring language, to the Macintosh.[29,30] The intention is that faculty members using this authoring system will find it much easier to develop high-quality, educational software for their classes. A further advantage of cT is its portability. It is now becoming possible to develop programs on Sun, VAXstation II, IBM AT and IBM RT workstations and have the software work on a standard Macintosh with little or no modification. By reducing programming time and allowing authors to make use of the latest features of a variety of machines, this new authoring system promises to make superior-quality, educational software available soon.

Trowbridge recently won the EDUCOM/NCRIPTAL award in both the physics and integrated categories for his *Graphs and Tracks* (see Fig. 7). This program allows students to go from motion to graphs and from graphs to motion. It has an extremely well-thought-out user-interface, which allows students to adjust the track (where the ball rolls) with simple mouse commands. Graphs are just as easily built using a palette of different graph segments, which are simply connected together and adjusted as desired. It is especially important to

Summary

Macintosh microcomputers bring new computing power to the physics laboratory. Many interesting and effective applications have already been produced by educators across the country. Ease of use and across-program consistency make these machines excellent tools for students in their investigations of the physical world. With the recent emergence of fast, efficient, software-development environments, we can look forward to the rapid introduction of even more low-cost, note that the entire package was based on the findings of physics education research carried out at the University of Washington. We need more software with a solid research foundation that makes use of proven pedagogic methods.

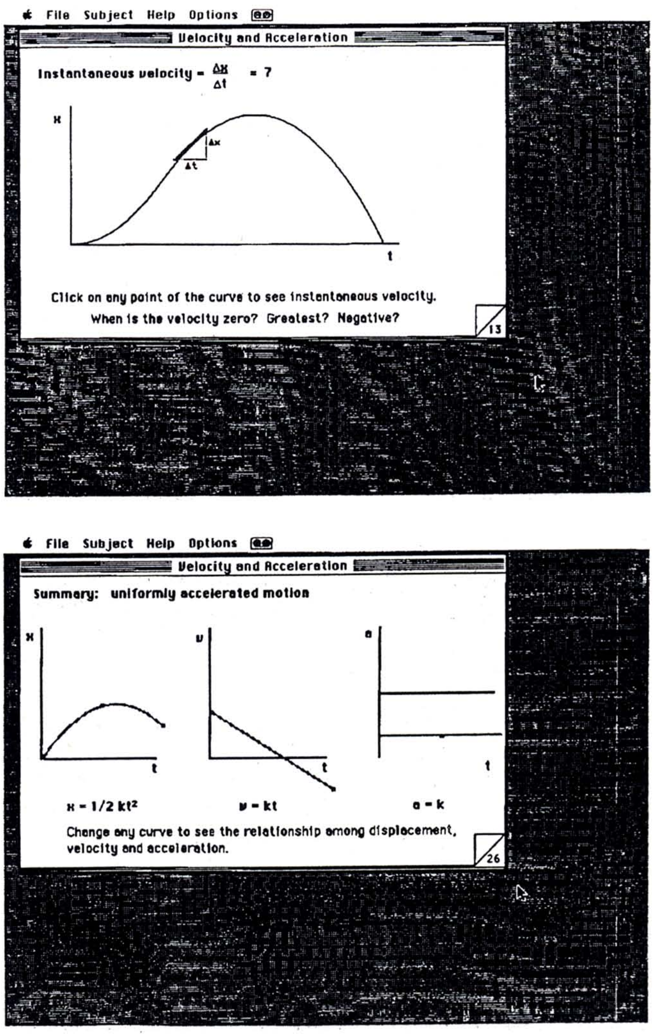


Fig. 6. Screens from the "Velocity and Acceleration" lesson in *Sensei Physics* by Broderbund Software. The package is a fairly extensive series of lessons, examples, and problems from nearly all areas of introductory physics.

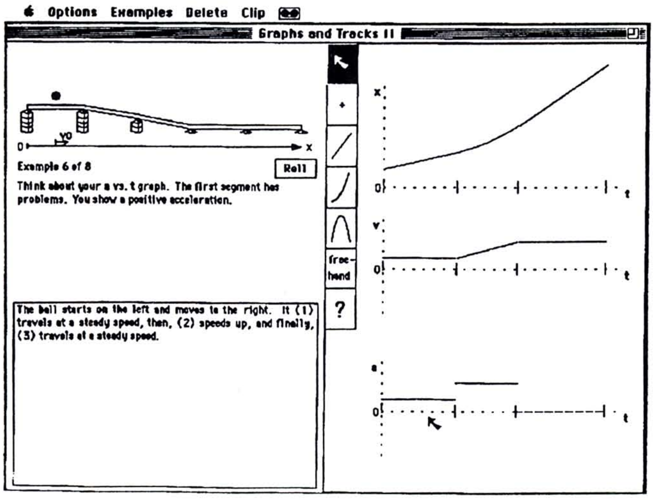


Fig. 7. Screen from *Graphs and Tracks* by David Trowbridge showing the MacDraw-like palette of graph segments, three student graphs, and the motion simulation area.

high-quality instructional software for the physics curriculum.

This article has attempted to briefly review the current state-of-the-art in CAI projects for the physics laboratory. All but one of these discussions have been specific to the Apple Macintosh microcomputer. I admit considerable bias on my part toward this particular machine, but I also found little information about what is being done with other computers of this caliber. It seems to me that most of the applications discussed here could be done as well on some other brand of computer with similar capabilities.

This presentation has by no means been exhaustive. Without a doubt, there are many interesting things going on that have been missed. I welcome information about these attempts (particularly those using Amiga, Atari, and the newer IBM computers) to improve the instruction taking place in our laboratory classrooms. I would also like to thank the many faculty members and equipment manufacturers who were willing to discuss their projects with me. I have included a longer than usual reference list so that readers can contact them for more information. These people are doing some very exciting things. I am sure they will be as helpful to you as they were to me.

References

1. Knaster, S., *How to Write Macintosh Software* (Hayden Publishing, Hasbrouck Heights, NJ, 1986), p. 139.
2. Pelkie, C. (Personal conversation at Cornell University, January 14, 1987) Division of Computer Services, Cornell University, Ithaca, NY 14852.
3. Risley, J. (Personal conversation at the Dickinson College Workshop on Computers in the Undergraduate Curriculum, January 16, 1987) Director, Software Evaluation Lab, North Carolina State University, Raleigh, NC 27695.
4. *Rascal*, available from Metaresearch, Pacwest Center Suite 2860, 1211 SW 5th Ave., Portland, OR 97204, 503-228-5806.
5. Taylor, E. (Letter to R. Beichner, February 10, 1987) Department of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139.
6. *MacApp and the Macintosh Programmer's Workshop*, available from Apple Programmer's and Developer's Association, 290 SW 43rd St., Renton, WA 98055, 206-251-6548.
7. *MacWorld* **4** (4) 115–117 (1987).
8. Contact: Peter Olivieri, *Wheels for the Mind*, Fulton 430, Boston College, MA 02167.
9. Contact: Academic Courseware Exchange, 4141 State St., Santa Barbara, CA 93110, 800-235-6919.
10. Berman, E., and Taylor, E., *Spacetime Software*, 20 Davis Rd., Belmont, MA 02178.
11. Fuller, R. (Letter to R. Beichner, March 6, 1987) Department of Physics and Astronomy, University of Nebraska, Lincoln, NE 68588.
12. Fuller, R., *Journal of College Science Teaching* **16**, 239 (1987).
13. For a review of the modelling software, see the Winter 1987 issue of TERC's newsletter *Hands On!* **10** (1) 18. *Stella* is available from High Performance Systems, Inc., 13 Dartmouth College Highway, Lyme, NH 03768, 603-795-4122.
14. Tinker, R., "Modeling and MBL: Software Tools for Science," *Proceedings of the National Educational Computing Conference*, June 4–6, 1986, San Diego, CA, p. 351.
15. Fuller, R., and Winch, D., Department of Physics, United States Air Force Academy, Colorado Springs, CO 80840.
16. Available from Metaresearch, Inc. Other interfaces include *MacADIOS* and *LabView*, discussed in "The Labtop Macintosh," *MacWorld* **3** (10) 136 (1986).
17. Osgood, D., "The Difference in Higher Education," *Byte* **12** (2) 165–178 (1987).
18. Elisha Huggins' package, called *MacScope*, is available from Metaresearch.
19. Thornton Associates, Inc., 1432 Main St. (Rte. 117), Waltham, MA 02154, 617-890-3399.
20. Contact: Special Projects Director, Office of Computing Services, Drexel University, Philadelphia, PA 19104. For a summary, see the Winter 1986 issue of *Wheels for the Mind* **4** (1) 228–229.
21. MacWorld News, *MacWorld* **4** (7) 103 (1987).
22. McGowan, C. (Personal conversation at the University of Rochester, January 6, 1987) Computer Center, University of Rochester, 727 Elmwood Ave., Rochester, NY 14620.
23. Zollman, D. (Letter to R. Beichner, February 25, 1987) Department of Physics, Kansas State University, Cardwell Hall, Manhattan, KA 66506.
24. Zollman, D., Noble, L., and Curtin, R., "Modelling the Motion of an Athlete: An Interactive Video Lesson for Teaching Physics" to be published in the *Journal of Educational Data Systems*.
25. Mazur, E. (Letter to R. Beichner, April 3, 1987) Division of Applied Sciences and Department of Physics, Pierce Hall 225, Harvard University, Cambridge, MA 02138.
26. MacWorld News, *MacWorld* **3**, 11, (1986).
27. Tinker, R., *Software Tools for Physics*. Paper presented at the conference on Computers in Physics Instruction, Raleigh, NC, August 1988.
28. Available from Broderbund Software, 17 Paul Dr., San Rafael, CA 94903-2101, 415-492-3555.
29. Anderson, B., and Trowbridge, D. (Personal conversation at the Dickinson College Workshop on Computers in the Undergraduate Curriculum, January 16, 1987) Center for Design of Educational Computing, UCC 20, Carnegie-Mellon University, Pittsburgh, PA 15213.
30. Trowbridge, D., Larkin, J., and Scheftic, C., "Design and Implementation of a Computer-Based Tutor," paper submitted to the 8th National Educational Computing Conference, NECC 1987.