

## **The Cosmos on a Computer: Computational JavaLabs for Astronomy Education**

*by Chris Mihos (CWRU) and Greg Bothun (U Oregon)*

Science is much more a method of inquiry into the nature of the physical world than it is a body of knowledge about it. In teaching science, we ideally want to involve students in the experimental process that lies at the core of scientific inquiry, and develop in them a scientific literacy that will enable them to better understand the scientific method and how science in general is conducted. Indeed, numerous studies have shown that collaborative learning methods and active experimentation are far better at conveying concepts than are traditional “fact-oriented” lectures.

However, teaching astronomy as an experimental science presents a unique set of challenges. The size scale of the universe means that astronomical objects are not easy to get one’s hands on, and even simply observing stars and galaxies is often impossible from urban settings. As a result, students often see astronomical objects as static objects – pictures in a textbook or on a website – and it remains difficult to connect students in an experimental way with modern astrophysics and cosmology.

This disconnect is unfortunate, because, like all sciences, progress in astronomy is driven by experimentation. Observationally, new data from ground- and space-based telescopes are motivating new theories in a number of areas, from the formation of planets to the expansion of the universe. Meanwhile, astronomers now use high speed supercomputers to model the detailed structure of stars, collisions of galaxies, and the evolution of structure in the universe. This synergy between observational data and theoretical modeling has resulted in an explosion of new ideas in astronomy and cosmology. Translating this experimental approach to students is difficult, because the computer experiments performed by research astronomers typically rely on specialized computer programs that are not straightforward for students to run, and the large datasets that result from both observational and computational experiments are often unwieldy for students to manipulate.

To address these issues, we have been developing a series of web-based computer exercises which allow students to work with computer modeling and large astronomical datasets in an experimental manner. The technical challenge associated with this task is to deliver the data and software to students while minimizing the need for specialized software or complex computer expertise. The most efficient way of providing the materials to the students is through the construction of Java “applets” — computer programs which run inside a web browser — which we then make available via the World Wide Web. Java is a programming language developed by Sun Microsystems that allows us to design sophisticated, interactive tools such as computer modeling or data visualization software, which students can use within the familiar context of the web browser environment.

Nearly all students have Java-capable browsers (i.e. Netscape or Internet Explorer) already installed on their computers and no additional software is required to run our JavaLabs. The browser environment is also good for integrating text and graphics that

can be used as supporting material for the Java-based exercises, providing all the products the students need in a single interface. An added bonus of the web interface is that, because the JavaLabs are publicly available over the web, anyone with a Java-capable web browser can experiment with the JavaLabs.

Java's power lies in its flexibility and our ability to integrate it into a web-based environment. The Java interface is far superior to that of traditional software products that students must install individually (and often pay for). The applets provide a portable means of providing the labs – students can access the labs anywhere, from any machine, and do not need to be working on a machine that has the lab “installed.” Similarly, because our servers host the JavaLabs, as opposed to the JavaLabs being resident on the students' computers, upgrades are instantly available, and the software distribution is uniform. As we bring more labs on-line, and upgrade existing ones, those updates are loaded onto our servers so that all users see the latest version as soon as it is available – there is no need for reinstalling a new version, and no worries that students may be using different, outdated versions. In practice, Java still suffers from some platform incompatibilities, but its overall combination of portability, standardization, and ease of upgrade is why we have chosen Java as the vehicle to deliver simulations, experiments, and data analysis tools.

Again, our principal goal with the JavaLabs is to provide students with an interactive and experimental scientific experience. A good example is the GalCrash applet (shown in Figure 1), in which students design their own computer models of colliding galaxies. Unlike the traditional view of galaxies as “island universes,” astronomers now believe that galaxies form and evolve hierarchically, through mergers with other galaxies large and small. Using the GalCrash applet, students actively experiment with galaxy collisions, learning about a variety of topics from galaxy evolution to gravitational dynamics.

We begin with an in-class discussion about gravity, galaxy interactions, and tidal forces, and the students learn the basic physics behind the process. Students then use the applet to test the physics. Setting up models of colliding galaxies, they can vary parameters such as the orbital geometry and encounter distance, as well as the structure of the galaxies involved. Once these parameters are set, the applet runs the computer model in real time, allowing the students to watch the collision evolve. By examining a variety of encounters, students explore what kinds of encounters give rise to tidal tails, and how differences in galaxy structure affect the outcome of the collision.

Perhaps the most popular use of the applet, however, is having students try to build models of real interacting galaxies like the Antennae or the Mice. By simulating these real galaxies, students can study the past history of the collision and, by running the simulation forward in time, the future fate of the galaxies. The JavaLab transforms the galaxies from static images taken at the telescope into an ongoing, evolving event that students witness on the computer.

In a second applet, RotCurve, students use rotation curves of real galaxies to study dark matter in spiral galaxies. The applet (Figure 2) plots the observed rotation curve, and students vary the amount of mass in the galaxy disk to try and match the observed

rotation speed of the galaxies. Quickly they realize that they need to add mass in the form of a dark matter halo in order to match the rotation curve; the applet lets the students choose the density and size of the dark halo to make this fit. The important point here is there is no unique solution — many different combinations of disk and halo properties can give reasonable fits to the data.

At first, this tends to frustrate students, who often expect that there is a single “right answer” they should be striving for. In fact, this degeneracy is a fundamental problem in our understanding of dark matter and spiral galaxies, and when students realize that they are dealing with the same problems faced in scientific research, they feel much more connected to astronomy as an ongoing scientific process.

Rotation curves exist for a variety of galaxies, allowing students to explore how the dark matter properties of galaxies differ as a function of galaxy type. These exercises mimic closely on-going astronomical research into the nature of dark matter. In an interesting twist, students in one class became involved in an argument as to how much of the inner velocities of galaxies should be supported by an unseen halo, much the same way that research astronomers argue whether or not the inner portions of disk galaxies are dominated by dark matter (the so-called “maximum disk” problem).

In yet another applet, students use libraries of real stellar spectra to study the spectral types of stars. Figure 3 shows a snapshot of the applet that compares the spectra of an A star (top) to that of a K giant (bottom). This applet has both quantitative and qualitative aspects. In its qualitative mode, the background of some elemental lines are shown (in this case, hydrogen) and the student can then try to identify and qualitatively assess the strengths of the hydrogen lines in stars of different spectral type. In the quantitative mode, the actual strength of the line (the “equivalent width”) can be measured; a comparison of the quantitative results produced by this applet with similar routines in the sophisticated data reduction packages used in astronomical research (such as IRAF and IDL) shows no disagreement. That is, the students are using a research grade tool and measuring real line strengths in real stars as a regular homework assignment in Astronomy 101.

After studying the spectral properties of stars, students can then use a new applet (seen in Figure 4) to combine different stellar spectra together to reproduce the integrated spectra of galaxies of different types – spirals, ellipticals, and some starburst galaxies. In essence, students are using spectroscopy to explore the mix of stars found in different types of galaxies — a process research astronomers call “population synthesis.” The applet gives students tools to measure the equivalent widths of spectral lines in the individual stars, the synthesized spectra, and the actual observed galaxy spectra. The applet also reports the relative contribution of each type of star to the visual and blue luminosity of the galaxy. This application has been used in elementary astronomy courses in mostly a qualitative manner. Its design, however, allows it to be used by more advanced classes and indeed, it has even proven useful at the graduate level. Moreover, all these applets can be used in a dynamic way in class presentations to illustrate key points.

Many other applets exist, aimed at a variety of educational levels from K12 through graduate school. Different JavaLabs focus on stellar orbits in galaxies, the expansion of the universe, the moons of Jupiter, the Hubble law, and basic physics such as atomic absorption and emission and the inverse square law of light. The learning curve for student use of the applets is often rather steep. Both students and educators need to be patient and to expect that a variety of different approaches will be needed to successfully use the applets. Again, this is how science is done: using trial and error to hone in on the best approach to a problem.

An essential ingredient for the successful use of the applets is a well defined lesson plan or tutorial that functionally integrates the applets (this is particularly important as one moves from simulation to actual data reduction, as in the case of our JAVA CCD simulators). Without this essential guidance from the instructor, the JavaLabs run the risk of being used as video games, where students push buttons and watch neat things happen, but get no real understanding of the underlying science involved. To provide both a contextual framework and a guide for using the JavaLab, the applets come embedded within a web environment that includes general scientific background information, an on-line manual for the applet, suggested exercises, and links to supplementary material.

A major design goal of our applets, and frankly one that has received mixed reactions, is that they do not use perfect data or present canned simulations. Instead, they consist of real data or active simulations complete with experimental noise; different students can run the applets and get different results based on how they interpret this noise. Students often claim that their results (for instance, the derivation of the inverse square law light curve using a very noisy detector) do not look like material that is presented in the book, or that their results differ from those of other groups. From the educator's perspective, this ambiguity gives a natural platform for discussing issues such as scientific uncertainty, the meaning of error bars, the concepts of inference versus proof, and the role of scientific debate. For students who are used to finding "the answer" in their homework problems, the JavaLab exercises can be a dramatic and, in some cases, even disturbing change in how they view science. Understanding that science is a process which is simultaneously rigorous and yet open to personal judgment and interpretation comes hard for many students. But by the end of the term we find that students have gained a much better appreciation for the process of science, from inquiry and hypothesis testing to data analysis and interpretation.

Indeed, that understanding is the fundamental educational goal we are trying to achieve through the use of these interactive, data-driven exercises. Astronomy is a naturally engaging subject to many students; as such, it provides an excellent opportunity to teach students – and in particular, non-science students – basic scientific literacy. Over the course of this project our own teaching goals have changed significantly. We consider it far more important for the student to finish the class with, for example, an appreciation of the role uncertainty plays in the scientific method, than having memorized the evolutionary track of the Sun in the HR diagram.

To realize these goals, we have had to both alter our teaching expectations and goals and build the requisite tool kit to engage the students in a new mode of learning: one that is actually based on real scientific inquiry. Moreover, JavaLab exercises can facilitate

collaborative learning among students (Figure 5), as different teams are assigned to tackle various aspects of the problem. For instance, in the GalCrash applet, some student teams are asked to determine which encounters best produce strong tidal tails, another team tries to create distorted spiral galaxies, while a third team studies which encounters lead to mergers. While this increases the work load compared to traditional textbook homework, it immerses the students in a collaborative, team-oriented learning environment. The skills they learn through this process – how to ask questions, how to test ideas, and how to apply concepts across a variety of problems – are skills which are valuable well beyond the classroom setting.

Finally, there is always the issue of whether students learn more by using these tools. Quantitative assessment of whether or not students “learn more” is hard – increased averages on exam scores are a rather indirect measurement of student understanding. What we do know is that students are being forced to learn in new ways, which, while initially frustrating, in the end gains them much more. To quote from a recent student evaluation:

*“This class was initially very frustrating for me as it seemed the instructor deliberately provided a confusing path to the answer. Over time, however, I realized that this was the only class I have taken in which I was actually doing science. For the first time, I now understand and appreciate that science is a process. Thanks!”*

Chris Mihos (hos@burro.astr.cwru.edu) is an Associate Professor of Astronomy and Physics at Case Western Reserve University in Cleveland, OH, where he studies how galaxies form and evolve. He is also a firm believer in loud music, professional football, comic books, microbreweries, and the wisdom of the common donkey.

Greg Bothun (nuts@bigmoo.uoregon.edu) earned his Ph.D. from the University of Washington in 1981 and has held research and teaching positions at Harvard, Caltech, the University of Michigan and the University of Oregon, where he has been since 1990 and is now a Professor of Physics. He prefers cows over donkeys.

### **JavaLab Websites**

Both the Case Western Reserve University and the University of Oregon JavaLabs are publicly available. The CWRU JavaLab can be accessed at <http://burro.astr.cwru.edu/JavaLab>, while the University of Oregon JavaLab is online at <http://jersey.uoregon.edu/vlab>. Funding for the JavaLabs has come from the National Science Foundation, Research Corporation, and Case Western Reserve University.