

# A New Approach to Developing Interactive Software Modules Through Graduate Education

Nathan E. Sanders · Chris Faesi · Alyssa A. Goodman

Published online: 27 October 2013  
© Springer Science+Business Media New York 2013

**Abstract** Educational technology has attained significant importance as a mechanism for supporting experiential learning of science concepts. However, the growth of this mechanism is limited by the significant time and technical expertise needed to develop such products, particularly in specialized fields of science. We sought to test whether interactive, educational, online software modules can be developed effectively by students as a curriculum component of an advanced science course. We discuss a set of 15 such modules developed by Harvard University graduate students to demonstrate various concepts related to astronomy and physics. Their successful development of these modules demonstrates that online software tools for education and outreach on specialized topics can be produced while simultaneously fulfilling project-based learning objectives. We describe a set of technologies suitable for module development and present in detail four examples of modules developed by the students. We offer recommendations for incorporating educational software development within a graduate curriculum and conclude by discussing the relevance of this novel approach to new online learning environments like edX.

**Keywords** Science education · Web tools · Simulations · Web-based learning · Demonstrations · Software development

## Introduction

Instructor demand for and utilization of educational technology to enable experiential learning is well-documented (see e.g., Bell et al. 2008; Campbell et al. 2010). Providing interactive software tools and manipulable simulations in addition to traditional printed textbooks can enhance the classroom and out-of-classroom learning experience for some students (see Vogel et al. 2006 for a meta-analysis and review). Modern educational resources include software modules for desktop (Sadler et al. 1999; Härtel 2000; Monaghan and Clement 2000; Gazit et al. 2005; Kumar and Sherwood 2007; Lamb and Annetta 2012) and mobile (Cheng and Tsai 2012; Powell and Mason 2013; van der Kolk et al. 2013) devices and programs or Web sites that enable distributed research or “citizen science” (Shell et al. 2011; Friedrich 2013) or bring results from academic research journals into the classroom (Sanders et al. 2012).

However, instructors typically lack the time and/or engineering expertise to develop software or online content of this type, and software developers often lack the content knowledge and/or interest to develop modules with significant educational value for advanced students. This has historically limited the availability of such software modules to only the most broadly salient topics, or content areas with high associated investment from industrial or academic research institutions.

We set out to test a novel approach to the development of specialized software modules, utilizing graduate students as a set of experts with the prerequisite programming experience and content knowledge. The module development task was integrated into a graduate course curriculum. Our motivations for enlisting graduate students in this effort are as follows.

First, the software development exercise is intended to fulfill the same educational objectives as other long-term

---

N. E. Sanders (✉) · C. Faesi · A. A. Goodman  
Department of Astronomy, Harvard University, 60 Garden St.,  
MS 42, Cambridge, MA 02138, USA  
e-mail: nsanders@fas.harvard.edu

project-based learning initiatives. These objectives include achieving student engagement and self-motivation, promoting conceptual understanding through problem-solving and decision-making in complex tasks, and appreciation for the application of scientific principles to realistic problems (Blumenfeld et al. 1991; Thomas 2000). Such projects may promote specialized learning through incorporated literature reviews and writing assignments. If instructors encourage interaction through in-class presentations or collaboration, the projects may also facilitate peer-learning.

Second, the product of the students' work is a publicly accessible and reusable educational tool. Such software modules are often not available for specialized topics frequently encountered by astrophysics educators at the advanced undergraduate and graduate levels. Graduate students in astrophysics can produce modules from their position as content experts, via their direct research experience and/or concurrent coursework. The student products have the potential to enhance learning on their target topics, as they may be the first interactive educational tools available for the subject. For topics with appeal to the broader public, the software products may serve an additional public outreach purpose. After a course ends, students, faculty, and technical professionals can collaborate to incorporate robust versions of relevant modules into formal "massive" online learning systems, currently represented by systems like edX.

Here, we present the results of our test, a set of new software modules developed by graduate students in the Harvard University Department of Astronomy in the Spring of 2013 as a term project for the AY201b graduate course, "The Interstellar Medium and Star Formation." Each of the fifteen enrolled students was required to produce a software module, amounting to a significant percentage of the total course grade. Project oversight was provided by the course professor (A. Goodman) and graduate teaching fellows (N.E. Sanders and C. Faesi). In "[Technologies for module development](#)" section, we discuss and compare the set of software technologies which the students used to develop their modules. We present a description of four example modules in "[Examples of student modules](#)" section. We conclude with a set of recommendations for instructors seeking to implement a similar module development class requirement ("[Recommendations for instructors](#)" section) and a discussion of how modules can be used in the context of massive online learning platforms ("[Evaluation and reuse of modules in cMOOC environments](#)" section).

## Technologies for Module Development

Various software technologies provide convenient frameworks for students to develop online interactive modules. The key features desired in these technologies are ease of

use (approachable learning curve for graduate students with research computing, but not software development, backgrounds), support (availability of sufficient tutorial material), suitability (preexisting capability to display astronomical imagery, interactive plotting, or other scientific data visualization), and accessibility to users (online playback in Web browsers on common platforms).

We have identified four technologies from the current software development landscape which best meet these criteria. Table 1 summarizes these technologies. In the following subsections, we describe each technology and provide Web links to support resources for students developing modules.

### WorldWide Telescope





WorldWide Telescope (WWT; <http://www.worldwidetelescope.org/Home.aspx>) is an interactive scientific data visualization and aggregation package developed by Microsoft Research. WWT includes a comprehensive variety of all-sky and high-resolution astronomical imagery from across the electromagnetic spectrum, as well as a 3D universe visualization that can be zoomed from earth to the solar system (animated with physically realistic orbital mechanics) to the large-scale structure of the universe (visualizing 3D galaxy positions from the Sloan Digital Sky Survey). Users can import their own catalog data or imagery in a wide variety of standard formats, including ASCII, VOTable, and Excel for tables, and JPEG and FITS for images.

A key feature of WWT is the ability to author interactive narrated tours (<http://www.worldwidetelescope.org/ExperienceIt/ExperienceIt.aspx?Page=Tours>).<sup>1</sup> The tours are built from a series of "slides," consisting of static or animated views of astronomical imagery. A tour proceeds by transitioning between these views, and it is typically accompanied by explanatory narration from the tour author. Users can pause the tour at any time during playback and manipulate the imagery displayed by the client (e.g., turning on overlays of observations in different wavelengths or rotating a 3D view of the solar system), and tours can include hyperlinks to Web resources with additional information.

The flagship client software for WWT is a free Microsoft Windows desktop software package. WWT Web clients have also been developed for both the Microsoft Silverlight plugin (already released at <http://www.worldwidetelescope.org/webclient/>) and for HTML5-enabled browsers (currently available as an application programming interface, API). At

<sup>1</sup> An example WWT tour that can be viewed on YouTube, without the WWT desktop client or the Silverlight Plugin, is available here: [https://www.youtube.com/watch?feature=player\\_embedded&v=3pKfGXMeUxg](https://www.youtube.com/watch?feature=player_embedded&v=3pKfGXMeUxg).

**Table 1** Summary comparison of suitability and technical requirements for authors and users for module development technologies discussed in this article

	 <b>Telescope</b>	 <b>JavaScript/ d3</b>	 <b>Wolfram CDF</b>	 <b>Python</b>
<b>Free to use</b> authoring software	Y	Y	N	Y
Likelihood of graduate student prior <b>familiarity</b>	Low	Low	Medium	Medium
<b>Technical experience</b> required for module development	Low	High	Medium	High
<b>Authoring</b> software platform	Windows	Windows, Mac, Linux and Unix variants	Windows, Mac, Linux	Windows, Mac, Linux and Unix variants
Software installation requirements for module <b>user</b>	Low	Low	Medium	High
Technical requirements for <b>distribution</b>	Low	High	Low	Medium

present, tours can only be authored in the desktop client, but tours can be played back in the desktop *or* Web clients.

Developing a tour in WWT requires the least technical/programming experience of any of the technologies described here. WWT has extensive online documentation (<http://www.worldwidetelescope.org/help/SupportHelp.aspx>), video tutorials (<https://wwtambassadors.org/wwt/video-tutorials>), as well as an active community of WWT outreach experts, the WWT Ambassadors program (WWTa), led by A. Goodman and collaborators in association with Microsoft Research, available as resources. The WWTa Web site (<http://wwtambassadors.org>) can provide hosting of and distribution of tours free of charge to authors, enabling tours to reach a large audience assuming it has passed the quality requirements imposed by the WWTa program.

WWT offers extensive capabilities to display imagery, but relatively little functionality to display other data types interactively. The remaining technologies in our survey offer more flexibility, but developing advanced visualizations requires significantly more technical expertise from authors.

#### JavaScript/d3

JavaScript is an interpreted programming (scripting) language. JavaScript is commonly used in so-called Web 2.0 platforms such as Google's Gmail and to perform smaller

interactive functions in a vast number of pages on the Web. JavaScript's functionality will be familiar to users of other scripting languages popular among graduate student researchers, such as Python. Several JavaScript programming tutorials are available online (see for example <http://www.codecademy.com/tracks/JavaScript>)

Many libraries have been developed and distributed to extend the core functionality of JavaScript. A popular JavaScript library for data visualization is d3.js (<http://d3js.org/>), standing for "Data-Driven Documents." d3 allows for the display of scientific data in a wide variety of formats, including line plots and bar charts and extending to network graph browsers and map projections.

Scores of example pages produced in d3 are available on the project Web page (<https://github.com/mbostock/d3/wiki/Gallery>), complete with full source code and data files. These examples can serve as templates for student module development. Additionally, Web pages are available that allow users to experiment with modifications to JavaScript and d3 code while watching the results appear in real time within a Web browser window (see for example [http://phrogz.net/JS/d3-playground/#BoxMullerDistribution\\_HTML](http://phrogz.net/JS/d3-playground/#BoxMullerDistribution_HTML)).

A prerequisite for students intending to develop modules using JavaScript is familiarity with other Web technologies, including HTML for Web page development, and access to a Web server for hosting (as many university departments provide).

## Wolfram Computable Documents

The Computable Document Format (CDF), designed by Wolfram Research (<http://www.wolfram.com/>), takes interactive code written in Wolfram's Mathematica software and renders it graphically in either a stand-alone player or via a browser plugin. Mathematica is a high-level computing language that is commonly used for a wide range of scientific and mathematical applications such as equation solving, numerical integration, two- and three-dimensional plotting, and data analysis. It will already be familiar to many graduate students through use in coursework and/or research and has a shallow learning curve and thorough tutorials and documentation. Mathematica is proprietary software that must be purchased, although institutional licenses are common, and a student (discounted) version is also available.

CDF relies on the Mathematica “manipulate” command, which allows plot inputs to be customized by the user via interactive elements such as sliders, radio buttons, and menus. Plotting is updated dynamically. Code is written directly in the Mathematica language and formatted appropriately for CDF viewing. Text can be incorporated to provide topical background and orient the user with the module. Importantly, the user need only have the CDF player to view the document, which is available as a free download.

The CDF platform has been utilized by Wolfram to launch the new Wolfram Demonstrations Project, a central repository of user-created modules in science, engineering, mathematics, and economics (<http://demonstrations.wolfram.com/>). Demonstration source codes are available for download and can serve as templates or examples. Student modules are ideally suited for submission to the Demonstrations Project, and one of our students, Lauren Woolsey, has already had her module on the Zeeman effect accepted to the Project (<http://demonstrations.wolfram.com/TheZeemanEffectInTheInterstellarMedium/>). The increased visibility possible through such promotion is an additional beneficial side effect of using the Wolfram CDF technology for module implementation.

## Python

Like JavaScript, Python (<http://Python.org/>) is an interpreted programming language. However, it is typically used for desktop application development or for server-side scripting rather than in Web applications run in the browser. A disadvantage of Python as a module development technology is therefore that it requires users to download and manually run software on their computers, and users must have already installed the Python framework. However, software developed in Python does not necessarily need to be installed by end users and can be run as a stand-alone ASCII script file (depending on what associated data files and libraries are

needed). The software can be hosted using any Web space or file hosting service. And (soon) when “hosted” web-ready Python installations come to “MOOC” platforms like edX, then learners using modules installed on those MOOC platforms will not need a desktop Python installation.

The advantage for module authors is that many graduate students are already familiar with Python and its associated scientific data analysis and visualization modules (numpy, <http://www.numpy.org/>, and matplotlib, <http://matplotlib.org/>) from their research experience. Like JavaScript, many online programming tutorials are available for learning Python (e.g., <http://www.codecademy.com/tracks/Python>).

The matplotlib library allows for the creation of sophisticated data visualizations using Python. Any matplotlib visualization can be made animated and/or interactive using an object-oriented programming interface for adding customizable buttons, switches, sliders, and other interface objects. As in the case of d3, code for plots from a library of examples (<http://matplotlib.org/examples/index.html>) can be used as templates for student projects.

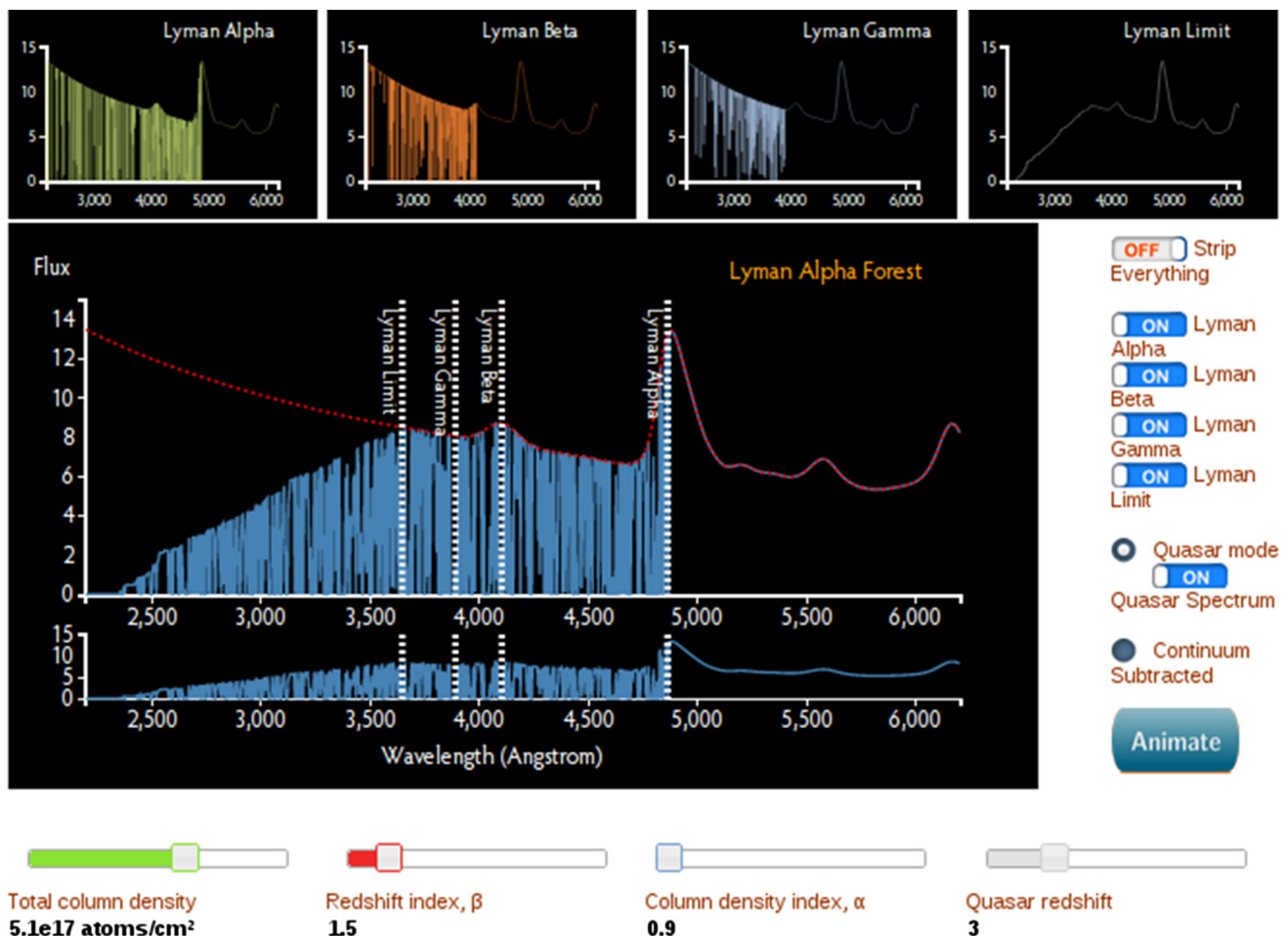
## Examples of Student Modules

Here, we describe in detail four of the fifteen modules created by graduate students in the Spring 2013 semester Harvard AY201b course. These modules have been selected to showcase the diversity of subject matter, software technologies, and target audiences addressed by the student projects. Summaries of and Web links to the remaining modules can be found at the AY201b course Web site (<http://ay201b.wordpress.com/topical-modules/2013-topical-modules/>).

### Simulating the Lyman Alpha Forest, Yuan-Sen Ting

This module demonstrates the physical origin of the Lyman Alpha forest absorption features seen in the optical spectra of high-redshift quasars, one of the most important observational probes of the early universe (see e.g., Rauch 1998 for a review).<sup>2</sup> The primary feature of the module (available at [https://www.cfa.harvard.edu/~yuan-sen.ting/lyman\\_alpha.html](https://www.cfa.harvard.edu/~yuan-sen.ting/lyman_alpha.html)) is a physically realistic and fully interactive visualization of the formation and observable spectrum of the Lyman Alpha forest, created using d3 and shown in Fig. 1. Interactive scalers and switches allow users to adjust the physical parameters of the simulated system, the properties of the displayed spectrum, and to animate the appearance of absorption features as a function of light-travel time.

<sup>2</sup> Ting presents additional discussion of the Lyman Alpha forest and the associated Gunn–Peterson effect in an article produced for a separate assignment for the AY201b class, available online at <http://ay201b.wordpress.com/2013/04/20/on-the-density-of-neutral-hydrogen-in-intergalactic-space-2/>.



**Fig. 1** Screenshot of the Lyman Alpha forest JavaScript module developed by Yuan-Sen Ting

Hovering over the interactive buttons yields pop-up boxes with the descriptions of their function.

The module is written in JavaScript and has several interactive features in addition to the simulation shown in Fig. 1. First, two versions of the detailed explanatory text (addressing of relevant topics in cosmology and spectroscopic line formation) are available for different audiences, (1) the general public and (2) undergraduate and graduate students in astrophysics.<sup>3</sup> A series of animated figures from the astrophysical literature accompany the advanced versions. Second, an additional interactive simulation (also made using d3) illustrates the formation of different spectroscopic line profiles as a function of physical conditions and the associated position on the curve of growth.

Ting's work exemplifies a full-featured interactive module suitable for use by several different audiences. The module supports inquiry-based exploration by individual students, who can use the simulation to test and

visualize hypotheses for the effects of changes to physical parameters that they generate based on descriptions they have read of Lyman Alpha forest formation in textbooks or Ting's own explanatory text. Classroom discussion at the undergraduate and graduate level can be facilitated by exploring similar hypotheses in a group setting, replacing the display of static plots or schematic chalkboard illustrations. Finally, as the simulation is scientifically accurate, it can be applied by research astronomers working in related fields to generate animations to accompany scientific presentations.

#### Landmarks of the Interstellar Medium, Meredith MacGregor

Meredith MacGregor's WorldWide Telescope Tour "Landmarks of the ISM" guides viewers through the broad range of phenomenology in the interstellar medium (ISM) of the Milky Way, stopping at specific well-known objects (landmarks) and discussing relevant physical processes. This includes a multi-wavelength view of Kepler's

<sup>3</sup> Dr. Steven Cranmer and Dr. Bruce A. Peterson, in consultation with Ting, contributed to the explanatory text for the graduate student version of the Web site.





**Fig. 2** Screenshot of the WorldWide Telescope interface with MacGregor's tour, "Landmarks of the ISM"

supernova remnant, an infrared window into newly formed stars in NGC 2264, and a description of ionization in the IC 1396A ("Elephant's trunk") emission nebula.

Developed using the WWT Windows desktop client, the tour is available as a 16 megabyte standalone file (hosted on the module Web site, <http://ismlandmarks.wordpress.com/>). The file includes the tour slides and playback program, audio narration recorded by the author and music, and images not already available through the WWT interface. The tour can be viewed by importing this file into the Windows desktop client or cross-platform Web clients (Fig. 2).

The tour and its narration are designed for the general public, particularly for viewers with exposure to science approximately at a high school level. The tour includes hyperlinks to pages (hosted on the module Web site) with additional details on topics discussed in the narration, including more advanced information and literature citations for learners at the undergraduate or graduate level.

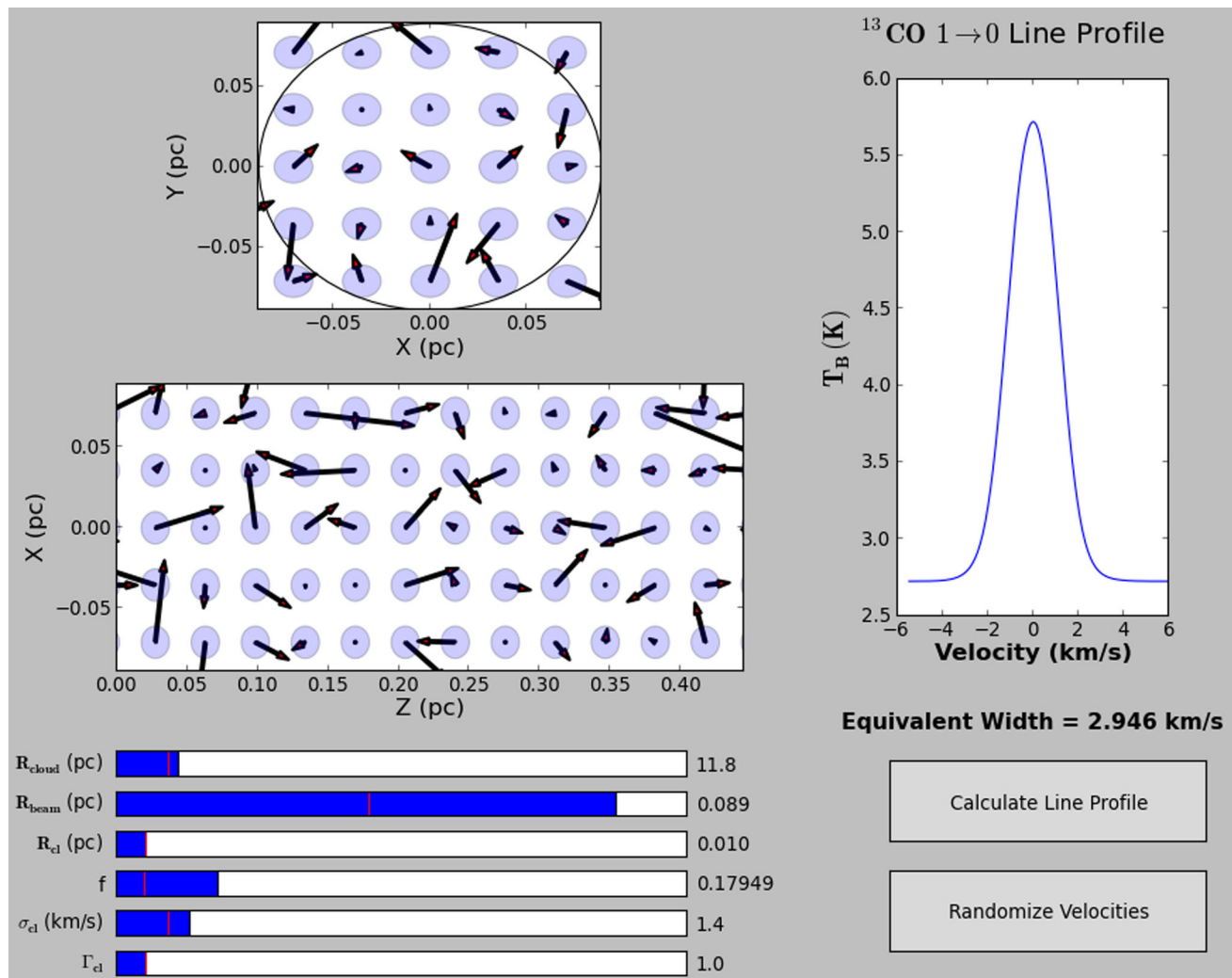
The Effects of Clumpiness in Giant Molecular Clouds,  
George Miller

In combination with its accompanying Web site (<http://clumpmodel.wordpress.com/>), this module enables users to

simulate the observational signature of clumpiness in the structure of giant molecular clouds, the sites of star formation within galaxies (see e.g., Wolfire et al. 1993). Users can vary physical parameters, including the clump size and filling factor, the width of the radio telescope beam, and the velocity dispersion of the clumps. Figures on the module interface visualize the clump geometry and velocity distribution, and a button allows users to simulate the observed line profile for their desired configuration (see Fig. 3).

The module was developed by George Miller using Python and the matplotlib library. The total codebase, including both simulation backend and user interface, consists of only about 300 lines of Python code distributed as a single script file. Users on any platform with Python installed can launch the code with a single command, with no installation necessary, as documented on the module Web site.

The accompanying explanatory material on the module Web site is appropriate for upper-level undergraduate or graduate students. Students at that level can use the module to explore the effects of different cloud parameters on the radio spectrum and build intuition needed to interpret observations. The module could be used to facilitate in-class discussions of structure in the interstellar medium, or as a resource to accompany laboratory exercises involving radio telescopes.



**Fig. 3** Screenshot of the interactive module visualizing the effect of clumpiness on the observed line profiles of molecular clouds, developed by George Miller

### Magnetohydrodynamic Shocks, Philip Mocz

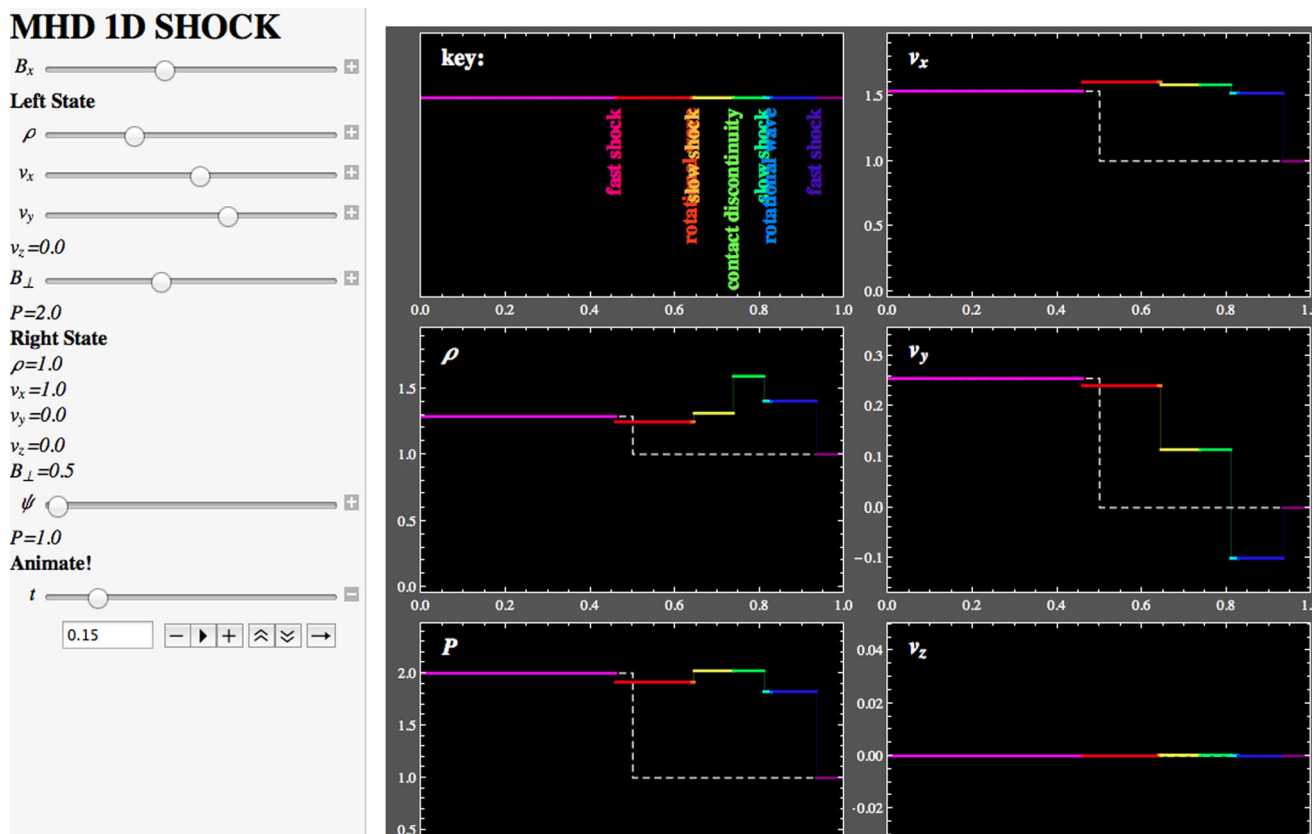
Philip Mocz's module (<https://www.cfa.harvard.edu/~pmocz/shocks/index.html>) provides an introduction to one-dimensional hydrodynamic shocks. The module allows users to explore the implications of the shock discontinuity for the fluid variables density, velocity, pressure, and magnetic field. Both the inviscid (zero viscosity) and magnetohydrodynamic (MHD) cases are included. The Web site includes a thorough introduction to the theory behind shocks (particularly the Riemann problem, e.g., Menikoff and Plohr 1989) as well as background material, describing how more complex and realistic three-dimensional problems can be solved using numerical methods.

In the module itself, the user is invited to adjust the values of the fluid variables on either side of the shock discontinuity and observe the effects on the structure of the

shock through dynamically updating plots (Fig. 4). The Web site also includes a helpful list of suggested adjustments for the user to try when first using the module. This provides a natural connection between the theory and the demonstration, and gives the user context and guidance in exploring the visualization.

This module is written in Mathematica and viewed in a Computable Document Format (CDF) player embedded directly in the Web site. The interactive portion is seamlessly integrated with the supplementary content. The user needs to only download the free CDF player in order to access the full set of online materials.

As with several other modules produced for AY201b, Mocz's work is suitable for upper-level undergraduate or graduate students. The background material is thorough in introducing the shock-specific information, but assumes some understanding of advanced mathematics. Liberal use



**Fig. 4** Screenshot of the Mathematica-based module on one-dimensional magnetohydrodynamic (MHD) shocks developed by Philip Mocz

of schematics and figures aids intuition, and the interactive module itself builds nicely upon the theory. Since shocks are ubiquitous in many areas of astronomy as well as in physics, this Web site could serve as a supplementary material for a range of courses, from fluid dynamics to the interstellar medium to high-energy astrophysics.

### Recommendations for Instructors

Here, we provide a set of recommendations for instructors who wish to implement a module development project as we have described in a graduate curriculum. These recommendations reflect our experience in teaching the Harvard AY201b course and are intended to provide support to the graduate student module developers and to improve the final products made available to users.

First, we recommend beginning the module development project by providing students with brief in-class tutorials of the software technologies to expose them to relevant choices and functionality. In our case, we spent a total of 2 h, spread over two class periods, on live tutorials, led by the Teaching Fellows. “Technologies for module development” section of this article includes links to a variety of online tutorial programs and sample/template

codes. The modules now available on the AY201b Web site (<http://ay201b.wordpress.com/category/special-topics-modules/>) can serve as further examples. For classes of students with limited past experience in computation, we recommend that instructors focus on using the module technologies identified as having lower technical experience requirements in our Table 1 (Worldwide Telescope and Wolfram CDF). Additionally, instructors should identify technologies relevant to the subject matter of their course with which their students may have prior familiarity. This may include field-specific technologies beyond those we present here in the context of astrophysics.

Second, we recommend regular consultations between students and the teaching staff to monitor the progression of module development. An initial proposal should be submitted by students for instructors to evaluate the intended scope of the project. Requiring students to give in-class presentations of their module well before the final due date can serve the dual purpose of peer-teaching (with the presentation including substantive description of the science topic and the module targets) and providing instructors with a snapshot of development progress so they may return feedback. Additional lecture discussion of the modules science target by instructors can ensure that the course’s curriculum goals are met. For about half of the



fifteen modules created in AY201b, substantial revisions to a student's module scheme were made after consultations and/or in-class presentations.

Some common difficulties may be experienced by students during module development. Technical barriers may halt progress; instructors may wish to encourage affected students to focus on the scientific content of the module and, meanwhile, help students find workarounds or contact other experts to troubleshoot technical issues. As well, students may have difficulty focusing their piece for a well-defined audience. If students have produced material too specialized or advanced for the intended target audience, this can be separated into a distinct online Web page (as in the Ting module) or, in the case of a WWT tour, a second version (a "Director's Cut") can be made that includes additional details. This second version can easily be produced in the WWT authoring tool as an alternate audio narration track.

Third, it is critical for end users that the software modules be accompanied by written explanatory material. This should serve as an instruction manual for the software, as well as a pedagogical exercise for the students in documenting the scientific concepts and their modules seeks to illustrate. Developing this text will play an important role in fulfilling the teaching objectives of the project, as students investigate their specialized topics in great detail and synthesize information as they would in a traditional term paper assignment. This material can be easily hosted online using a free service such as WordPress.com, which Ting and MacGregor have done (see "Examples of student modules" section).

Finally, we strongly recommend that students make a narrated video tour of the software available on an associated Web page. This video will ideally show users the key functionality of the software module and also discuss the most important scientific concepts it illustrates. For an example of such a narrated video tour, see the one produced by Ting for his Lyman Alpha forest module ([http://www.youtube.com/watch?v=v1Dj92-rX\\_8](http://www.youtube.com/watch?v=v1Dj92-rX_8)). Such tours are particularly valuable for modules made using technologies not native to the web (e.g., Python)

### Evaluation and Reuse of Modules in cMOOC Environments

We conclude with a discussion of the importance of these student-developed modules in the broader context of the growing landscape of online course materials, and a path forward for evaluating the effectiveness of the modules as educational tools. The students who created the AY201b modules were exceptionally motivated and worked intensely on their projects. The key motivator for the students

was the idea that the work they were doing on a term project would be of legitimate use for others trying to learn the material they sought to explain.

Student writing and software products from the AY201b course have been made publicly available online through several means. Students in both the 2011 and 2013 versions of the biennial course contributed to Web page style content on specific topics as their term projects. The AY201b WordPress site (<http://ay201b.wordpress.com>) that incorporated the student content receives a surprisingly large number of views (>10,000 per year) given the specialized nature of its content. Moreover, the 2013 AY201b graduate students were aware that their online modules would *all* be linked to the WordPress site and that *some* modules would also be selected for incorporation into edX (see <http://edX.org>), the Massive Online Open Course ("MOOC") environment founded by MIT and Harvard in 2012. The selection of which modules went to edX would be (and now is being) made based on a combination of how relevant the module's subject matter is to advanced undergraduate courses, the usability of the module ("user experience"), and how adaptable the module's technology is to a Web-based platform (see Table 1).

edX, at present, is offering primarily lecture-based courses, in their entirety, online. That form of MOOC, where an end-to-end replica of an in-person course is made available online, is known as an "xMOOC." An alternative form of MOOC is known as a "cMOOC," where the "c" stands for "connectivist" (see <http://degreeoffreedom.org/xmooc-vs-cmooc/>). One key goal of the AY201b module experiment was to see if graduate students could create material suitable for inclusion in a cMOOC-like environment. Now that the answer appears to be "yes, graduate students can produce very valuable cMOOC content," our team is working with the edX staff at Harvard (HarvardX) to pair students whose modules are most suitable for a broad audience with technical professionals within the HarvardX organization, so that they can recast the modules into forms that will withstand high-user demand online. It is likely that the three WWT-based modules developed will be put online first, thanks to WWT's visual appeal and the relatively introductory-level nature of the modules made using it.

Interest in expanding HarvardX to include cMOOC-like modular content has been brewing ever since edX was announced, and it appears that the AY201b modules will be the first cMOOC-style content added to edX. The intent is twofold: (1) Learners seeking to know more about the specific topics covered will discover the modules on their own, much as they would otherwise discover a whole course on a broader topic; and (2) teachers at all levels seeking multimedia content, to incorporate into both "live" courses and online ones, will find these modules and use

them in much the same way they would have used articles or videos discovered in libraries or online in the past.

An important ancillary benefit of the inclusion of these modules in a cMOOC environment is the potential to use data from this system for evaluation. The edX platform allows for detailed monitoring of who uses what material and how, including the data needed to search for correlations between usage of each module and individual student performance across all courses throughout the edX environment. By exploring this data, we will be able to evaluate the educational impact of these modules using a large sample of online student users. We plan to report these usage and evaluation results in a follow-up publication after sufficient data have been collected, as well as to evaluate the impact of integrating these modules into future classroom versions of the Harvard AY201b course.

The interactive student-generated content we discuss here could easily be implemented in graduate level courses across the sciences. While assignments of this type may be most easily implemented in fields where knowledge of computer programming is prerequisite, they may hold even more educational value for students in fields where it is not. The finished online modules can serve as useful educational tools for all disciplines.

**Acknowledgments** We are grateful to all the students of the AY201b course for their participation in this course project and for their excellent module products, and to HarvardX Director Prof. Robert Lue for his support of the teaching staff of the course. We thank Kate Alexander and Yuan-Sen Ting for helpful comments. A.G. thanks Katie Vale for her expertise on MOOCs and educational technology, and Jonathan Fay at Microsoft Research for his constant development and support of the WWT program. N.E.S. and C.F. are supported by the National Science Foundation through Graduate Research Fellowships.

## References

- Bell R, Gess-Newsome J, Luft J (2008) Technology in the secondary science classroom. NSTA Press, Arlington
- Blumenfeld PC, Soloway E, Marx RW, Krajcik JS, Guzdial M, Palincsar A (1991) Motivating project-based learning: sustaining the doing, supporting the learning. *Educ Psychol* 26(3&4):369–398
- Campbell T, Kwei Wang S, Hsu H-Y, Duffy AM, Wolf PG (2010) Learning with web tools, simulations, and other technologies in science classrooms. *J Sci Educ Technol* 19:505–511
- Cheng K, Tsai C (2012) Affordances of augmented reality in science learning: suggestions for future research. *J Sci Educ Technol* 22(4):449–462
- Friedrich JM (2013) A classroom-based distributed workflow initiative for the early involvement of undergraduate students in scientific research. *J Sci Educ Technol*. doi:10.1007/s10956-013-9450-z
- Gazit E, Yair Y, Chen D (2005) Emerging conceptual understanding of complex astronomical phenomena by using a virtual solar system. *J Sci Educ Technol* 14(5–6):459–470
- Härtel H (2000) xyZET: a simulation program for physics teaching. *J Sci Educ Technol* 9(3):275–286
- Kumar DD, Sherwood RD (2007) Effect of a problem based simulation on the conceptual understanding of undergraduate science education students. *J Sci Educ Technol* 16(3):239–246
- Lamb RL, Annetta L (2012) The use of online modules and the effect on student outcomes in a high school chemistry class. *J Sci Educ Technol* 22(5):603–613
- Menikoff R, Plohr BJ (1989) The Riemann problem for fluid flow of real materials. *Rev Mod Phys* 61:75–130
- Monaghan JM, Clement KJ (2000) Algorithms, visualization, and mental models: high school students' interactions with a relative motion simulation. *J Sci Educ Technol* 9(4):311–325
- Powell CB, Mason DS (2013) Effectiveness of podcasts delivered on mobile devices as a support for student learning during general chemistry laboratories. *J Sci Educ Technol* 22(2):148–170
- Rauch M (1998) The Lyman alpha forest in the spectra of quasistellar objects. *Annu Rev Astron Astrophys* 36:276–316
- Sadler PM, Whitney CA, Shore L, Deutsch F (1999) Visualization and representation of physical systems: wavemaker as an aid to conceptualizing wave phenomena. *J Sci Educ Technol* 8(3):197–209
- Sanders N, Kohler S, Newton E, the Astrobites Collaboration (2012) Preparing undergraduates for research careers: using astrobites in the classroom. *Astron Educ Rev* 11:1
- Shell DF, Snow GR, Claes DR (2011) The cosmic ray observatory project: results of a summer high-school student, teacher, university scientist partnership using a capstone research experience. *J Sci Educ Technol* 20(2):161–177
- Thomas JW (2000) A review of research on project-based learning. [http://www.bie.org/index.php/site/RE/pbl\\_research/29](http://www.bie.org/index.php/site/RE/pbl_research/29)
- van der Kolk K, Hartog R, Beldman G, Gruppen H (2013) Exploring the potential of smartphones and tablets for performance support in food chemistry laboratory classes. *J Sci Educ Technol*, March 2013
- Vogel JJ, Vogel DS, Cannon-Bowers J, Bowers CA, Muse K, Wright M (2006) Computer gaming and interactive simulations for learning: a meta-analysis. *J Educ Comput Res* 34:3
- Wolfire MG, Hollenbach D, Tielens AGM (1993) CO(J = 1-0) line emission from giant molecular clouds. *Astrophys J* 402:195