

Interactive Worksheets for Teaching Atmospheric Aerosols and Cloud Physics

Markus Petters

ABSTRACT: Student-centered active learning pedagogies improve learning outcomes and increase the engagement of underrepresented groups. Implementing such pedagogies requires interactive tools for students to manipulate inputs and use the outputs to construct knowledge. This work introduces interactive worksheets for teaching about atmospheric aerosol and cloud physics and describes the toolchain to create and deliver the content. The material is appropriate for upper-level undergraduate and graduate instruction with pedagogy based on process-oriented guided inquiry learning. Students playfully interact with physical relationships and atmospheric models. Two examples are the interaction with an aerosol–cloud parcel model for simulating the early stage of cloud formation and the interaction with the Bowen model for simulating the formation of rain by coalescence. Photos, text, figures, and software associated with the project are free to be shared and free to be adapted. In addition to focusing on discipline-based learning objectives, the worksheets emphasize interacting with real-world data and practicing graph comprehension. Hosting the content in the cloud ensures reliable and scalable delivery to any device with a browser and Internet access. The worksheets are designed to be used in a student-centered active learning classroom but can also be used in an online setting.

KEYWORDS: Aerosols; Cloud droplets; Cloud microphysics; Drop size distribution; Measurements; Aerosol-cloud interaction

<https://doi.org/10.1175/BAMS-D-20-0072.1>

Corresponding author: Markus Petters, mdpetter@ncsu.edu

Supplemental material: <https://doi.org/10.1175/BAMS-D-20-0072.2>

In final form 12 August 2020

©2021 American Meteorological Society

For information regarding reuse of this content and general copyright information, consult the [AMS Copyright Policy](#).

AFFILIATION: Petters—Department of Marine, Earth, and Atmospheric Sciences, North Carolina State University, Raleigh, North Carolina

Active learning pedagogies are becoming more popular in higher education. Example short active learning activities include think-pair share exercises, brainstorming exercises, and minute papers. More formalized pedagogical approaches include flipped or blended classrooms (Baepler et al. 2014), problem-based learning, process-oriented guided inquiry learning (POGIL), and peer-led team learning (Eberlein et al. 2008). Active learning is student-centered. Collaborative teams work on specially designed activities. The instructor serves as facilitator while keeping formal lecture time to a minimum. Student-centered learning improves performance (Freeman et al. 2014; Soltis et al. 2015) and increases the engagement of underrepresented groups (Haak et al. 2011; Snyder et al. 2016).

Significant barriers prevent widespread adoption of student-centered pedagogies. Student-centered learning requires redesigned learning spaces, which are becoming available on many campuses but are not yet ubiquitous (Baepler et al. 2014). Few activity collections are available. For example, the POGIL project (www.pogil.org) lists POGIL-based textbooks in a variety of fields. These skew heavily toward chemistry and introductory courses. No texts are available for atmospheric science courses. Creating activities is time consuming for the instructor. Deslauriers et al. (2019) show that the increased cognitive effort associated with active learning leads to an incorrect student perception that they learn less. This can explain lower scores on student evaluations of teaching in active learning settings (Walker et al. 2008). All of the above conspire to limit adoption of student-centered approaches in environments where tenure and promotion criteria incentivize maximizing research productivity and numerical scores from student evaluations of teaching.

This work aims to lower the barrier to apply active learning in upper-level undergraduate and graduate courses. To this end, I introduce interactive worksheets for teaching about atmospheric aerosol and cloud physics. Applets embedded in the worksheets allow students to playfully interact with physical relationships and atmospheric models. Special emphasis is placed on incorporating real-world data and on practicing graph comprehension. The applets are built using emerging software technologies that allow scalable and reliable delivery of advanced content to any device with a web browser and Internet access. All content is free to be adopted, shared, and adapted. The framework is also suitable for archiving data and computing environments in compliance with new policies for scientific data management and stewardship.

Interactive worksheets

The interactive worksheets are delivered as Jupyter notebooks running on a cloud server (Perkel 2018). A Jupyter notebook is served to the user in a web browser and is organized into cells. A cell can display content written in the markdown markup language, which is similar to and simpler than HTML. These cells can display formatted text, formatted equations, videos, images, tables, and hyperlinks. A cell can also execute computer code written in programming languages such as Julia, Python, R, and many more. Output of the program can be placed in a cell and can consist of text or graphics. Using reactive programming (Elliott and Hudak 1997; Czaplicki and Chong 2013) one can create content that allows users to interact with the program through intuitive web elements such as range sliders, toggle buttons, switches, and input boxes. Figure 1 shows an example applet illustrating the vertical profiles of temperature, relative humidity, and liquid water mixing ratio during closed-parcel adiabatic ascent of an air parcel. Toggle buttons can be used to change the surface temperature and dewpoint temperature. Upon a click, the three graphs update to the new initial conditions.

The underlying thermodynamic relationships that govern the profiles in Fig. 1 are typically taught in a sophomore-level course on atmospheric thermodynamics, which is an informal prerequisite for the atmospheric physics course taught by the author. The example is used as refresher activity. During lecture, the activity is briefly introduced. Additional background information is provided in a cell that precedes the applet. Students then work in groups to

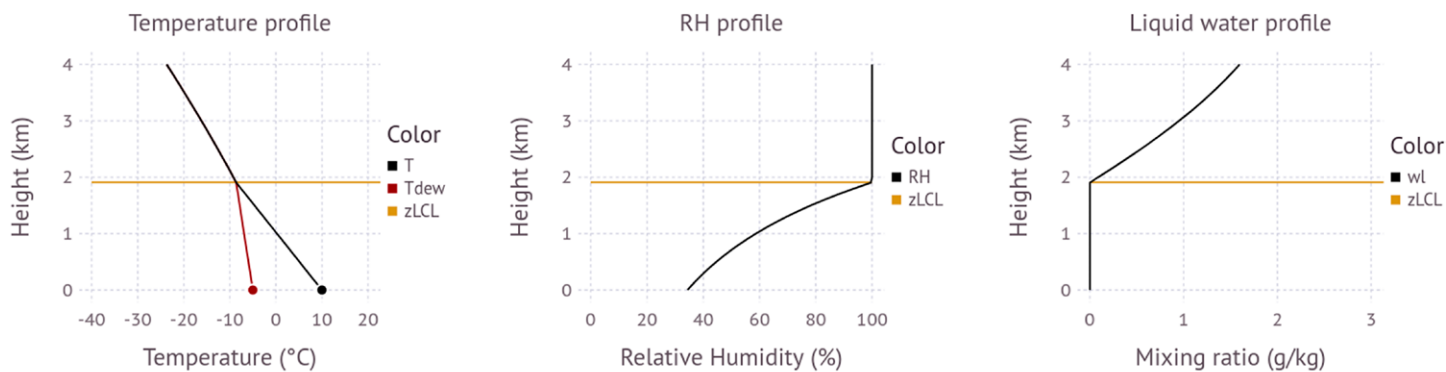


Fig. 1. Example applet produced using code running in a Jupyter notebook. LCL denotes the lifting condensation level, z height, RH relative humidity, and w_l the liquid water mixing ratio. The toggle buttons change the temperature (T) and dewpoint temperature (T_{dew}) at the surface.

answer guided inquiry questions. They interact with the applet by changing inputs and with their peers to cooperatively answer the questions. Individualized instruction supports the learning process. Upon completion of the activity, the instructor solicits answers to the questions from each group, followed by a summary of key points that are relevant to complete subsequent activities. Three example key points derived from this activity are that the height of the lifting condensation level (LCL) is determined by the difference between surface temperature and dewpoint temperature, that the increase in RH with height is nonlinear far below the LCL and approximately linear near the LCL, and that the liquid water mixing ratio scales with the LCL temperature and the height above the LCL.

The interactive worksheets are organized into modules. Currently eight modules are available, titled “Aerosol Dynamics,” “Cloud Condensation Nuclei,” “Early Stages of Cloud Formation,” “Cloud Droplet Size Distributions,” “Droplet Growth by Condensation,” “Droplet Growth by Collision and Coalescence,” “Influence of Aerosol on Precipitation,” and “Raindrop Size Distributions.” Modules on ice microphysics and radiation are planned to be released in the future. Learning objectives are articulated at the beginning of each module, followed by a sequence of explanatory content, applets, and questions. Explanatory content within the notebooks is limited to key points and equations and supplemented by minilectures, slides, and printed handouts. A single module is completed between one and four class periods.

Pedagogy

Overarching themes associated with the project content are the incorporation of real-world data and the focus on graph comprehension. Graph comprehension refers to the ability of students to derive meaning from graphs in a timely manner. Shah and Hoeffner (2002) reviewed the graph comprehension literature and made four recommendations. These are 1) to teach graph literacy skills in the context of science, 2) to translate representations to link visual features to the associated quantitative information, 3) to focus on the links between visual features and meaning, and 4) to train viewers to think of graph reading as an interpretation and evaluation task. Eye-tracking studies have shown that inexperienced viewers rely on cues in question prompts to complete the graph-based tasks, while experienced viewers focus on contextual elements such as graph titles, captions, and legends (Harsh et al. 2019). Frequent graphing assignments and a consistent graphical language help students to master graph comprehension (Shah and Hoeffner 2002; Violin and Forster 2018).

Formal design guidelines for graphical displays of quantitative information have been articulated by Kosslyn (1989), Wilkinson (1999), and Tufte (2001). The *Grammar of Graphics* introduced by Wilkinson (1999) inspired the plotting packages ggplot2 for the R language (Wickham 2016), and Gadfly for the Julia language (Jones et al. 2018). Graphical displays in the applets follow the *Grammar of Graphics* rules. By relying on a formal grammar, the contextual elements such as graph titles, construction of axis, and the use of a color key and/or a shape key are applied uniformly throughout. Furthermore, the *Grammar of Graphics* rules place graphical elements in a prescribed manner and restrict the creator by forbidding confusing constructs such as dual axes. This ensures that a consistent graphical language is used throughout and thus reduces the mental overhead associated with decoding graphs.

Figure 2 shows a typical example activity involving real-world data analysis while practicing graph comprehension. As suggested by Twomey (1959), cloud condensation nuclei (CCN) spectra are often suitably fit to a power law of the form $N_{\text{CCN}} = Cs^k$, where C and k are empirical coefficients and s is the supersaturation in percent. Adjustment of the range sliders moves the fit line. Students are tasked to tabulate C and k for the eastern Pacific, eastern Atlantic, Southern Ocean, and coastal Florida for continental and marine aerosol, which are coded by a color key and a shape key. They must decode the graph and associate model parameters with real-world data. Coefficient C corresponds to the CCN concentration at 1% supersaturation. By abstracting typical C values for marine and continental conditions students learn to link the visual features in the graph to the physical meaning related to aerosol loading.

The pedagogy is inspired by POGIL as described in Spencer (1999). Learning is modeled by the learning cycle which is backed by cognitive science and occurs in three phases. Phase 1 consists of exploration and data collection. Phase 2 consists of pattern identification and concept invention. Phase 3 consists of application, prediction and hypothesis testing. The activity shown in Fig. 2 captures phases 1 and 2. Phase 3, application and prediction, occurs in subsequent activities where students relate C and k to cloud droplet concentrations. However, there are also some differences compared to a dogmatic implementation of POGIL. In POGIL groups, distinct roles of manager, recorder, reflector, technician, and presenter are assigned (Eberlein et al. 2008). Lectures are almost completely eliminated, thus requiring that all knowledge is discovered and constructed. This proves difficult in an interdisciplinary atmospheric physics class where concepts from aerosol science, general atmospheric science, atmospheric thermodynamics, atmospheric chemistry, statistics, and general physics intersect. Thus minilectures are used to set the stage and group reporting to the class is used to ensure that all students are able to make the key connections.

Technical aspects

Language and implementation. The applets are programmed in the Julia language (Bezanson et al. 2017). Julia is a hybrid between interpreted languages such as Python and R and compiled languages such as C and Fortran. Julia code is compiled to native machine code using a just-in-time compiler during run time. Julia code execution speed is similar to that of

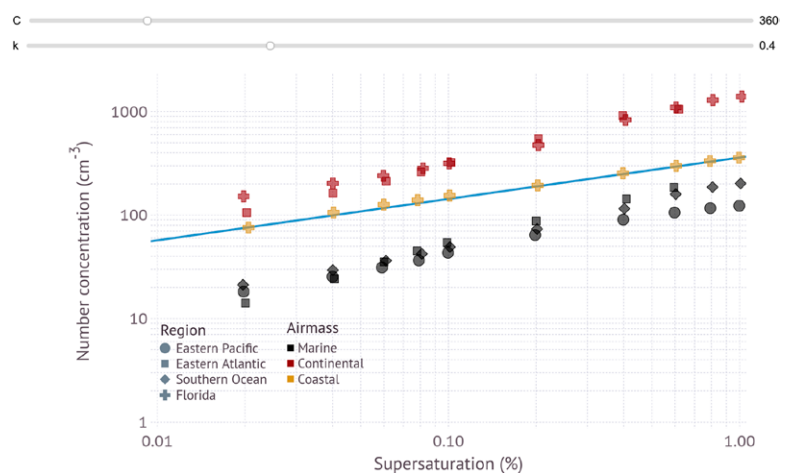


Fig. 2. Example applet showing the relationship between marine and continental CCN concentration as a function of supersaturation. Data are from Hudson and Yum (2001). The blue solid line corresponds to the fit $N_{\text{CCN}} = Cs^k$, where C (cm^{-3}) and k (nondimensional) are set by the range sliders above the graph.

strongly typed compiled languages, while the syntax and expressive power is similar to that of dynamically typed high-level languages like Python, R, MATLAB, and Lisp. The Julia code is executed in the Jupyter notebook. External packages provide key functionality: Interact.jl provides the reactive framework (Czaplicki and Chong 2013; Elliott and Hudak 1997), Gadfly.jl provides the *Grammar of Graphics* implementation (Wilkinson 1999), Compose.jl provides declarative vector graphics to create animations, and IJulia.jl provides Jupyter integration.

It is possible to call Python, R, C, or Fortran code from a Julia script. This functionality can be used to link numerical models written in a different programming language into an interactive applet. Figure 3 shows an example. The applet simulates the supersaturation profile as function of height during the early stage of cloud formation using an adiabatic parcel model (e.g., Rogers and Yau 1989). A parcel model solves the supersaturation balance equation in an updraft for a given aerosol. The parcel model code used here is written in Python/Cython and available for public use (Rothenberg and Wang 2016). The applet takes inputs of updraft velocity and aerosol number concentration. Other inputs needed to initialize a parcel model simulation, the aerosol size distribution, aerosol hygroscopicity parameter, thermodynamic state, and accommodation coefficient are assumed. This information is irrelevant for the exercise and hidden to the students. Here the focus is on the supersaturation generation term due to adiabatic expansion ($s = \alpha_1 z$, where s is the supersaturation, α_1 is a constant that models the supersaturation that is generated by the adiabatic expansion, and z is height above the LCL), the influence of updraft velocity and aerosol number concentration on the supersaturation profile $s(z)$, the maximum supersaturation s_{\max} , and maximum cloud droplet number concentration that is reached at heights exceeding the supersaturation maximum. Output from the simulation is parsed by the Julia script and displayed to the user. The update of the profiles is near instantaneous due to the Cython implementation of the parcel mode code. Thus the applet enables students to seamlessly interact with a numerical model written in a different programming language. Since there are few barriers between Julia and other code, this approach can be extended to almost any model that terminates fast enough to warrant creating an interactive applet.

Julia also has excellent support covering the gamut of numerical computing including, for example, optimization (Mogensen and Riseth 2018), machine learning

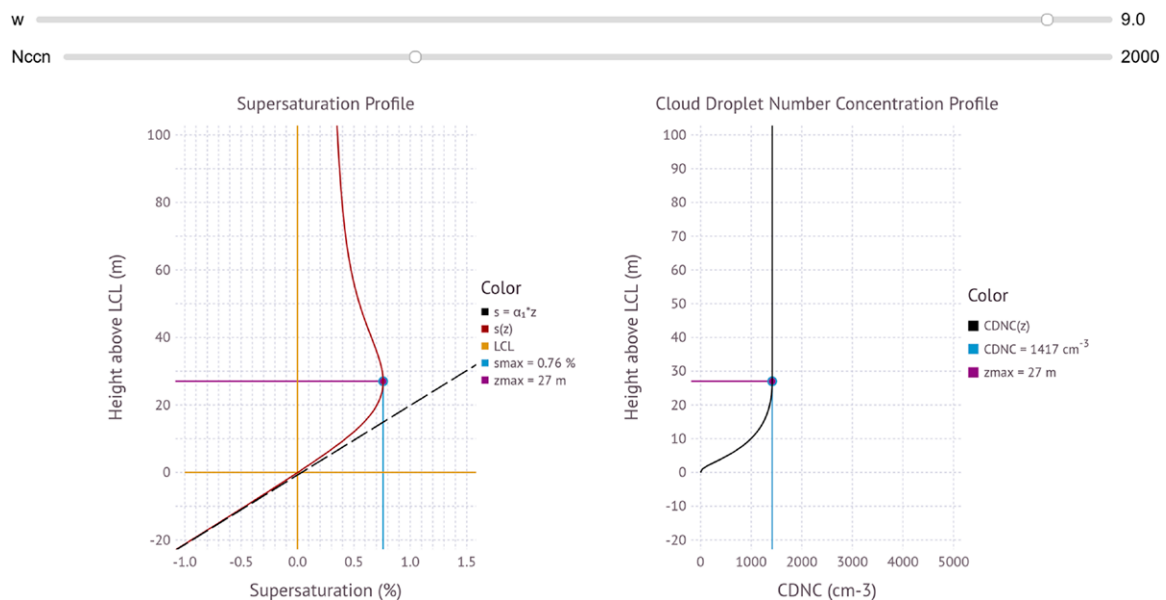


Fig. 3. Applet showing (left) the supersaturation profile and (right) the cloud droplet number concentration (CDNC) simulated from an adiabatic parcel model. The range sliders can be used to vary updraft velocity (w ; m s^{-1}) and aerosol number concentration (N_{CCN} ; cm^{-3}). The color legend is explained in the text.

(Malmaud and White 2018), Gaussian process modeling (Fairbrother et al. 2018), and differential equations (Rackauckas and Nie 2017). These libraries allow for quick implementation of educational models. One example included in the worksheets is the Bowen model describing the formation of rain by coalescence (Bowen 1950). It is often used to teach about the continuous collection equation and included as interactive applet where students can explore the effect of cloud droplet diameter, cloud liquid water content, and updraft velocity on the time to onset of rain and rain droplet size with height in convective clouds. The expressive power of the Julia language coupled with the extensive support for numerical computing enables fast execution of simple numerical models written in only a few lines of code.

Animations can also be created. Figure 4 shows an example animation illustrating the interaction volume for gravitational collection, where a collector drop of radius R grazes on cloud droplets with radius r . Drops begin to settle at their terminal velocity after switching the start-toggle. Settling on the screen is proportional to the terminal velocity. Students can change the radii of the drops and the spacing between the drops. Upon collision the applet creates a coalescence event and the drops merge. Inquiry questions guide students to derive the mathematical expression of the mass and radial growth rates due to gravitational collection. Functional reactive animation (Elliott and Hudak 1997) is used to program the applet.

Deployment. Installing the computational environment required to execute the notebooks can be challenging. Therefore a Docker container is used to bundle and serve the software. Docker containers provide operating system level virtualization. They can be executed on Windows, Macintosh, or Linux, but require a working install of the freely available Docker engine software. Executing the container starts the notebook server, which is then accessed through a web browser on the host machine. It is possible to view and change the source code inside the container. However, these changes are lost once the session is terminated. This ensures that the user always starts with a fresh, and thus working copy of the applets.

For seamless deployment in classes, the container can be served using a Kubernetes cluster hosted by a cloud computing provider. Costs of approximately \$30 per student and semester must be covered to pay for the cloud infrastructure. JupyterHub for Kubernetes is freely available software for container distribution. The Kubernetes cluster creates and outward facing IP address that can be linked to a university or private domain name. Users authenticate on the hub through a web browser. Each user is

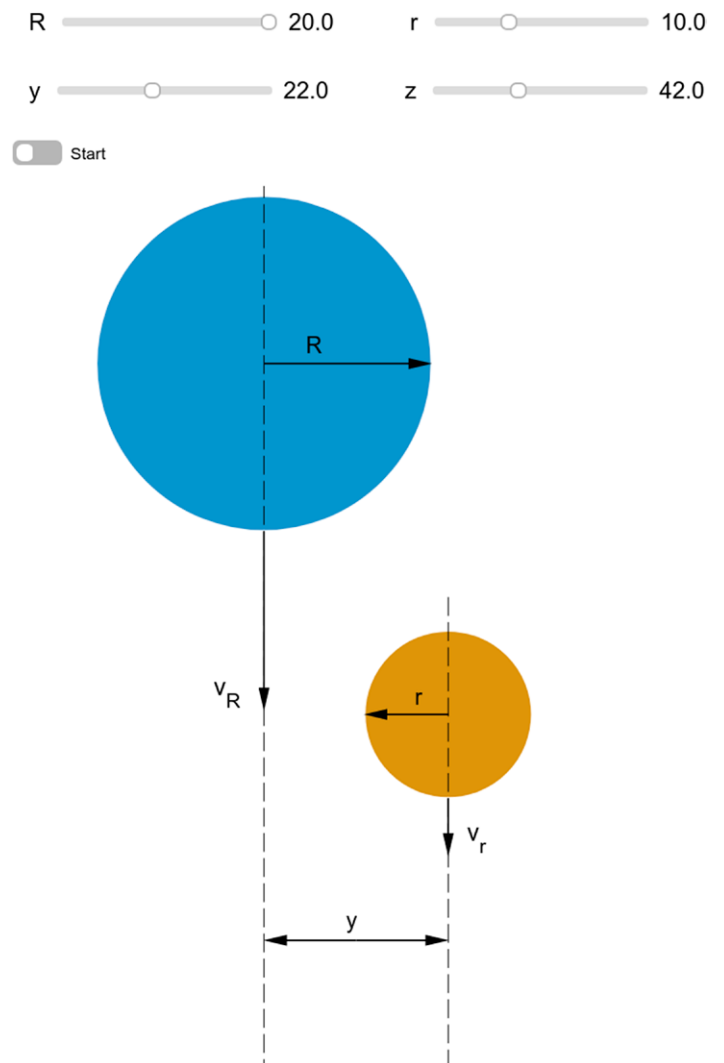


Fig. 4. Applet animating a collector droplet with radius R grazing on droplets with size r . The drops settle at terminal velocity v_R and v_r , respectively. The range sliders can be used to set the droplet radii, the horizontal distance between drop centers (y) and the vertical distance between drop centers (z). All units are in μm .

served their own instance of the Docker container image. The cluster scales with demand and can handle hundreds of instances simultaneously to serve large classes. Serving the software through a cloud computing provider has several advantages. Computer expertise to use the notebooks is not required. The notebooks are accessible through any device with a web browser, including mobile phones and tablets. Finally, cloud services are generally reliable allowing instructors and students to focus on teaching and learning, and not on system administration. Additional technical information is provided in the online supplemental information (<https://doi.org/10.1175/BAMS-D-20-0072.2>).

Discussion and conclusions

The material herein adds to a growing list of active learning approaches adopted by the meteorology community. Raymond (2007) introduced an active learning course in atmospheric chemistry and physics serving predominately chemical engineering majors at the undergraduate level. Davenport (2019) introduced the worked examples pedagogy, which guides students through assessments of real-world examples to improve learning in atmospheric dynamics. Steeneveld and Vilà-Guerau de Arellano (2019) describe an active learning course to teach numerical weather prediction at the M.S. level. Tanamachi et al. (2020) present an active learning field course in field observations in and near severe storms. Petters (2018) introduced Jupyter notebooks to create noninteractive tutorial materials associated with otherwise difficult-to-access scientific computing algorithms, which then inspired this work. Undoubtedly many more unpublished active learning courses are being taught.

This work outlines new content in the form of interactive applets and the technical approach to create and deliver them. The presented tools lower the barrier of entry to use active learning in atmospheric physics at the junior and senior level. The interactive worksheets have been used in three courses in 2019 and 2020. The approach was assessed through anonymous written student feedback reported in detail in the supporting information. Eight anonymous written responses were obtained. Six out of eight responses included positive feedback using phrases such as “liked the worksheets,” “valuable for understanding,” “loved the format/the interactive website was amazing,” “exciting new learning tool,” “the best way the class like this could be taught,” and “enjoyed the modules.” One student found the worksheets “adequate” but also an “odd form of learning.” One student would have preferred shorter modules. Three students commented that they wished for more instruction time and two students struggled with the approach to have to “figure it out on our own.” The assessment shows that using the interactive worksheets has a positive effect on student engagement. Students enjoy the activities and recommend using it. However, it is also true that students are poorly equipped to assess their own learning (Kruger and Dunning 1999; Stark and Freishtat 2014; Deslauriers et al. 2019). Students may enjoy the format and not learn as well as they would have in a traditional setting. Conversely, students that struggle with the cognitive load to “figure it out yourself” may not enjoy it but still learn more than they would have in a traditional setting. The course taught by the author is offered infrequently and has low enrollment relative to introductory classes, where sections offered in parallel can be used to conduct formal assessments. Thus no claim can be made that the worksheet approach improves learning outcomes. Future studies assessing this approach are needed.

The worksheets are ideally delivered in a physical space that supports student-centered learning. Common characteristics of such spaces are a seat arrangement that supports group work, all students having access to a laptop computer, and each group having access to additional learning technologies such as whiteboards and projectors (Baepler et al. 2014). It is also possible to use the worksheets in an online setting with appropriate video-conferencing software. In such a setting, students are assigned to breakout rooms containing three to four students. Groups work on the activity in the same manner as in a student-centered classroom.

The instructor rotates through breakout rooms and groups can call the instructor through electronic means. Students collaborate through video chat, electronic whiteboards, and screen sharing. Screen sharing works well since students can demonstrate and explain how they manipulate inputs, while the other students in the room can follow along using their own instance of the applets.

An important component of the project is that it is open to be adopted, shared, and adapted. Illustrations and photos included in the notebooks are limited to content with free licenses, e.g., the creative commons attribution license. The content created by the author is provided through free licenses. All of the software tools used to create the worksheets, i.e., Julia, Julia packages, Jupyter, JupyterHub, and Docker, are distributed with free software licenses. Text-book costs cause financial hardship on students. Open licensed material can alleviate this hardship. Educational content must be continually adapted to the context of the instructional setting. Place-based instructional models engage students in activities that directly relate to their environment (Hallar et al. 2011). For this reason, the worksheets incorporate many measurements close to the author's home institution. Instructors who adopt the worksheets at other institutions should change—and thus must be permitted to change—the data to match their location and adjust activities to address their individual students' needs.

The worksheets presented here can be used “as is” in other courses. Modifying explanatory content, modifying the guided inquiry questions, rearranging the sequence of applets, or adding new applets can be achieved with relatively little effort. The software toolchain can also be used as a template to create new interactive content for other courses. The approach might be especially suitable to create a course in computing in the atmospheric sciences, because a fully curated computing environment is served. (This was the original motivation behind the development of JupyterHub.) Importantly, the interactive applets could be written using Python coupled with “Bokeh” or R coupled with “Shiny” and delivered through Jupyter/Docker/JupyterHub as described here.

Finally, the infrastructure presented here is also suitable to enable open science and to satisfy new data policies. Many current data policies call for archiving data and software in public repositories (supporting information). Nonfree-licensed computing environments are fundamentally in conflict with open science. They hide the work behind a paywall and the software to run the code may not be archived itself. However, even code written with only free software can be challenging to archive. Irreproducible computing environments can arise from unsatisfiable package dependencies caused by rolling software updates. The Docker virtualization as used here solves this problem by bundling the data, computing environment, model source code, model executable, and processing scripts into a stateless and operating-system independent image. The approach also includes machine-readable instructions to recreate the environment at a future date if need. Docker images can be archived alongside the source code in Findable, Accessible, Interoperable, Reusable (FAIR)-aligned data repositories (Wilkinson et al. 2016).

Acknowledgments. I thank Prof. Sandra Yuter for discussions about active learning pedagogies. The content is based on the textbooks by Rogers and Yau (1989) and Hinds (1999), *Twelve Lectures in Cloud Physics* by Prof. Bjorn Stevens, and definitions given in the electronic version of the second edition of the *Glossary of Meteorology*. Costs to test the cloud deployment in a class setting were covered by educational and technology fees levied by NC State University.

Data availability statement. The source code is available through GitHub (<https://github.com/mdpetters/Atmospheric-Physics-Notebooks>). The docker image is available through DockerHub (<https://hub.docker.com/r/mdpetters/atmospheric-physics-notebooks>). The source code and docker image are permanently archived at zenodo.org (<http://doi.org/10.5281/zenodo.3977570>). Instructions how to access these resources is described in detail in the supplemental information and the linked pages.

References

- Baepler, P., J. Walker, and M. Driessen, 2014: It's not about seat time: Blending, flipping, and efficiency in active learning classrooms. *Comput. Educ.*, **78**, 227–236, <https://doi.org/10.1016/j.compedu.2014.06.006>.
- Bezanson, J., A. Edelman, S. Karpinski, and V. B. Shah, 2017: Julia: A fresh approach to numerical computing. *SIAM Rev.*, **59**, 65–98, <https://doi.org/10.1137/141000671>.
- Bowen, E. G., 1950: The formation of rain by coalescence. *Aust. J. Sci. Res.*, **3A**, 193, <https://doi.org/10.1071/PH500193>.
- Czaplicki, E., and S. Chong, 2013: Asynchronous functional reactive programming for GUIs. *Proc. 34th ACM SIGPLAN Conf. on Programming Language Design and Implementation*, New York, NY, ACM, 411–422.
- Davenport, C. E., 2019: Using worked examples to improve student understanding of atmospheric dynamics. *Bull. Amer. Meteor. Soc.*, **100**, 1653–1664, <https://doi.org/10.1175/BAMS-D-18-0226.1>.
- Deslauriers, L., L. S. McCarty, K. Miller, K. Callaghan, and G. Kestin, 2019: Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. *Proc. Natl. Acad. Sci. USA*, **116**, 19251–19257, <https://doi.org/10.1073/pnas.1821936116>.
- Eberlein, T., J. Kampmeier, V. Minderhout, R. S. Moog, T. Platt, P. Varma-Nelson, and H. B. White, 2008: Pedagogies of engagement in science. *Biochem. Mol. Biol. Educ.*, **36**, 262–273, <https://doi.org/10.1002/bmb.20204>.
- Elliott, C., and P. Hudak, 1997: Functional reactive animation. *Proc. Second ACM SIGPLAN Int. Conf. on Functional Programming*, Amsterdam, Netherlands, ACM, 263–273, <https://doi.org/10.1145/258948.258973>.
- Fairbrother, J., C. Nemeth, M. Rischard, J. Brea, and T. Pinder, 2018: Gaussian-Processes.jl: A nonparametric bayes package for the Julia language. arXiv, <https://arxiv.org/abs/1812.09064>.
- Freeman, S., S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, 2014: Active learning increases student performance in science, engineering, and mathematics. *Proc. Natl. Acad. Sci. USA*, **111**, 8410–8415, <https://doi.org/10.1073/pnas.1319030111>.
- Haak, D. C., J. HilleRisLambers, E. Pitre, and S. Freeman 2011: Increased structure and active learning reduce the achievement gap in introductory biology. *Science*, **332**, 1213–1216, <https://doi.org/10.1126/science.1204820>.
- Hallar, A. G., I. B. McCubbin, and J. M. Wright, 2011: CHANGE: A place-based curriculum for understanding climate change at Storm Peak Laboratory, Colorado. *Bull. Amer. Meteor. Soc.*, **92**, 909–918, <https://doi.org/10.1175/2011BAMS3026.1>.
- Harsh, J. A., M. Campillo, C. Murray, C. Myers, J. Nguyen, and A. V. Maltese, 2019: “Seeing” data like an expert: An eye-tracking study using graphical data representations. *CBE Life Sci. Educ.*, **18**, ar32, <https://doi.org/10.1187/cbe.18-06-0102>.
- Hinds, W. C., 1999: *Aerosol Technology: Properties, Behavior, and Measurement of Airborne Particles*. 2nd ed. John Wiley and Sons, 483 pp.
- Hudson, J. G., and S. S. Yum, 2001: Maritime–continental drizzle contrasts in small cumuli. *J. Atmos. Sci.*, **58**, 915–926, [https://doi.org/10.1175/1520-0469\(2001\)058<0915:MCDCIS>2.0.CO;2](https://doi.org/10.1175/1520-0469(2001)058<0915:MCDCIS>2.0.CO;2).
- Jones, D. C., and Coauthors, 2018: *Giovineltalia/Gadfly.jl: V0.7.0*. Zenodo, <https://doi.org/10.5281/zenodo.1284282>.
- Kosslyn, S. M., 1989: Understanding charts and graphs. *Appl. Cognit. Psychol.*, **3**, 185–225, <https://doi.org/10.1002/acp.2350030302>.
- Kruger, J., and D. Dunning, 1999: Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments. *J. Pers. Soc. Psychol.*, **77**, 1121–1134, <https://doi.org/10.1037/0022-3514.77.6.1121>.
- Malmaud, J., and L. White, 2018: TensorFlow.jl: An idiomatic Julia front end for TensorFlow. *J. Open Source Software*, **3**, 1002, <https://doi.org/10.21105/joss.01002>.
- Mogensen, P. K., and A. N. Riseth, 2018: Optim: A mathematical optimization package for Julia. *J. Open Source Software*, **3**, 615, <https://doi.org/10.21105/joss.00615>.
- Perkel, J. M., 2018: Why Jupyter is data scientists' computational notebook of choice. *Nature*, **563**, 145–146, <https://doi.org/10.1038/d41586-018-07196-1>.
- Petters, M. D., 2018: A language to simplify computation of differential mobility analyzer response functions. *Aerosol Sci. Technol.*, **52**, 1437–1451, <https://doi.org/10.1080/02786826.2018.1530724>.
- Rackauckas, C., and Q. Nie, 2017: DifferentialEquations.jl—A performant and feature-rich ecosystem for solving differential equations in Julia. *J. Open Res. Software*, **5**, 15, <https://doi.org/10.5334/jors.151>.
- Raymond, T., 2007: Non traditional courses for applying STEM knowledge. *ASEE Conf.*, Honolulu, HI, American Society of Engineering Education, AC 2007-438.
- Rogers, R. R., and M. K. Yau, 1989: *A Short Course in Cloud Physics*. 3rd ed. Butterworth-Heinemann, 290 pp.
- Rothenberg, D., and C. Wang, 2016: Metamodeling of droplet activation for global climate models. *J. Atmos. Sci.*, **73**, 1255–1272, <https://doi.org/10.1175/JAS-D-15-0223.1>.
- Shah, P., and J. Hoefner, 2002: Review of graph comprehension research: Implications for instruction. *Educ. Psychol. Rev.*, **14**, 47–69, <https://doi.org/10.1023/A:1013180410169>.
- Snyder, J. J., J. D. Sloane, R. D. P. Dunk, and J. R. Wiles, 2016: Peer-led team learning helps minority students succeed. *PLOS Biol.*, **14**, e1002398, <https://doi.org/10.1371/journal.pbio.1002398>.
- Soltis, R., N. Verlinden, N. Kruger, A. Carroll, and T. Trumbo, 2015: Process-oriented guided inquiry learning strategy enhances students' higher level thinking skills in a pharmaceutical sciences course. *Amer. J. Pharm. Educ.*, **79**, 11, <https://doi.org/10.5688/ajpe79111>.
- Spencer, J. N., 1999: New directions in teaching chemistry: A philosophical and pedagogical basis. *J. Chem. Educ.*, **76**, 566, <https://doi.org/10.1021/ed076p566>.
- Stark, P. B., and R. Freishtat, 2014: An evaluation of course evaluations. *ScienceOpen*, <https://doi.org/10.14293/S2199-1006.1.SOR-EDU.AOFRQA.v1>.
- Steenefeld, G.-J., and J. Vilà-Guerau de Arellano, 2019: Teaching atmospheric modeling at the graduate level: 15 years of using mesoscale models as educational tools in an active learning environment. *Bull. Amer. Meteor. Soc.*, **100**, 2157–2174, <https://doi.org/10.1175/BAMS-D-17-0166.1>.
- Tanamachi, R. L., D. T. Dawson, and L. C. Parker, 2020: Students of Purdue Observing Tornadoic Thunderstorms for Research (SPOTTR): A severe storms field work course at Purdue University. *Bull. Amer. Meteor. Soc.*, **101**, E847–E868, <https://doi.org/10.1175/BAMS-D-19-0025.1>.
- Tufte, E. R., 2001: *Visual Display of Quantitative Information*. 2nd ed. Graphics Press, 197 pp.
- Twomey, S., 1959: The nuclei of natural cloud formation part II: The supersaturation in natural clouds and the variation of cloud droplet concentration. *Geofis. Pura Appl.*, **43**, 243–249, <https://doi.org/10.1007/BF01993560>.
- Violin, C. R., and B. M. Forster, 2018: An introductory module and experiments to improve the graphing skills of non-science majors. *J. Microbiol. Bio. Educ.*, **20**, 20.3.59, <https://doi.org/10.1128/jmbe.v20i3.1863>.
- Walker, J. D., S. H. Cotner, P. M. Baepler, and M. D. Decker, 2008: A delicate balance: Integrating active learning into a large lecture course. *CBE Life Sci. Educ.*, **7**, 361–367, <https://doi.org/10.1187/cbe.08-02-0004>.
- Wickham, H., 2016: *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, 266 pp.
- Wilkinson, L., 1999: *The Grammar of Graphics*. 2nd ed. Springer, 408 pp.
- Wilkinson, M. D., and Coauthors, 2016: The FAIR guiding principles for scientific data management and stewardship. *Sci. Data*, **3**, 160018, <https://doi.org/10.1038/sdata.2016.18>.