Climate Change: Science and Software

Steve M. Easterbrook, University of Toronto

Paul N. Edwards, University of Michigan

Venkatramani Balaji, Princeton University

Reinhard Budich, Max Planck Institute for Meteorology

CLIMATE CHANGE IS likely to be one of the defining global issues of the 21st century. The past decade—the hottest in recorded history—has witnessed countries around the world struggling to deal with drought, heat waves, and extreme weather. On current emissions paths, climate models project even more serious impacts on public health, agricultural production, fresh water supplies, extreme weather events, sea-level rise, ocean acidification, and fragile ecosystems.[1] The sheer scale of the problem makes it hard to understand, predict, and solve.

A Growing Field

Climate science relies on a vast software infrastructure that allows large teams of scientists to construct very complex models out of many interlocking parts and encourages scientists, activists, and policymakers to share data, explore scenarios, and validate assumptions.[2] The extent of this infrastructure is often invisible (as infrastructure often is, until it breaks down), both to those who rely on it and to interested observers such as politicians, journalists, and the general public. Yet weaknesses in this software (whether real or imaginary) will impede our ability to make much progress on the twin challenges of mitigation and adaptation to climate change.

As far as we know, no software magazine or journal has tackled climate software in a special issue. In planning this one for *IEEE Software*, we drew upon recent explorations of software challenges in computational sciences taken more broadly. For example, since 2008, a series of workshops on software engineering for computational science and engineering has included some discussion of climate science and numerical weather prediction, largely though comparison with other fields. This community produced two special issues of *IEEE Software* (July/August 2008 and January/February 2009) and a special issue of *Computing in Science & Engineering* (November/December 2009).

The geosciences community has also taken a growing interest in software and data handling for climate science via regular sessions at the larger geosciences conferences on topics such as model interoperability and software development strategy. Climate science journals regularly publish special issues on specific climate models, typically timed to present results from a major new release of a given model. However, these tend to focus on the new science that the model enables, rather than to describe the software and its development.

Climate Software

Technical discussions within the geosciences community about algorithms, numerical recipes, and scientific validation of the resulting models are often too narrowly focused to offer useful insights for the software community. Because climate change is a problem

0740-7459/11/$26.00 © 2011 IEEE

of major societal importance, software professionals have many critical roles to play in ensuring that the infrastructure on which this enterprise rests is built using the very best available tools and techniques.

Our motivation for this special issue, then, has been to help link the larger software community with the scientists, policymakers, and other stakeholders who currently develop and use climate software. We solicited articles covering any aspect of this software infrastructure, including the simulation models that drive our understanding of Earth system processes, the data-handling tools used to curate and analyze the huge volumes of observational data and simulation outputs, the assessment models used to study the impacts of policy choices, and the visualization and education tools used to explain these issues to wider audiences. Twenty articles were submitted—many more than we had hoped—each reviewed by at least three people representing a mix of expertise from both climate science and software engineering.

Based on the reviewers' recommendations, over half the submitted articles deserved publication, but due to page limits, we could select only four for this issue, leaving us in the uncomfortable position of having to omit some positively reviewed articles. To rectify this situation, we anticipate

- a follow-up special section of *IEEE Software* featuring several articles that describe the software challenges in handling petabyte-sized datasets in climate science (to appear sometime in 2012); and
- a special issue of the journal *Geoscientific Model Development* featuring a groups of articles that describe software challenges specific to the upcoming "Fifth Assessment Report of the Intergovernmental Panel on Climate Change" (which should also appear sometime in 2012).

Clearly, this isn't the last time you'll read about this important topic!

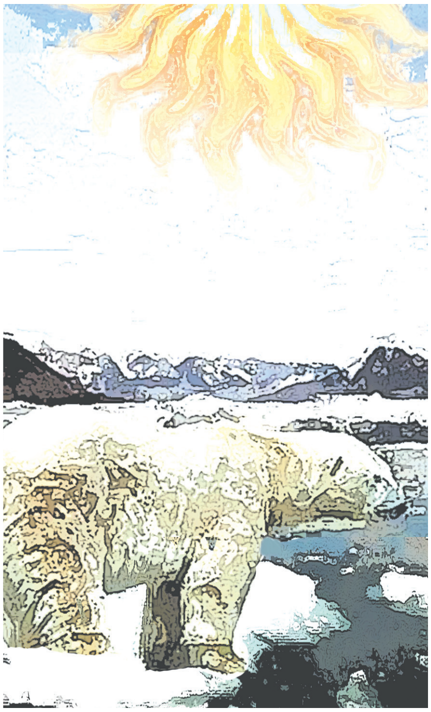
In This Issue

Climate science is a strongly computational discipline. It has always pushed the boundaries of computational feasibility, from the first numerical weather model (built as a challenging demonstrator project for ENIAC[3]) to the supercomputing facilities operated by modern climate research centers. The newest generation of Earth System Models (ESMs) strain the capacity of massively parallel supercomputing architectures, as they pass terabytes of data between processes.

Climate science is also strongly rooted in the collection and analysis of observational data. For this, climate scientists tap into a vast global data collection enterprise that includes data from satellites, ground stations, ocean buoys, and weather balloons. Most of this data network was initially designed only to support short-term weather forecasting. As a result, it often fails to fulfill the climate science requirement for long-term continuity and global coverage. Climate scientists also need data unrelated to weather, such as concentrations of greenhouse gases, which must be obtained from dedicated instruments or proxy sources. Adding to some of the challenges in the field, raw weather data tend to be very messy. Over the long periods (decades to centuries) required by climate science, weather stations change instruments, locations, operators, and standards, resulting in discontinuities that must be discovered and corrected. Another problem is that station locations don't map easily onto the regular grids used in climate software. As Paul Edwards argues in his book *A Vast Machine*,[2] the traditional distinction between observational data and scientific models has become (necessarily) blurred in both meteorology and climate science because raw data can't be used until they're processed through analysis models that take into account properties of the instruments, physical laws, and known adjustments to data sources. Consequently, data analysis is itself a major area of climate research.

The Nick Barnes and David Jones article in this issue, "Clear Climate Code: Rewriting Legacy Science Software for Clarity," reports on a case study in which one of the more prominent tools for processing observational data, GISTEMP, was rewritten in Python to produce cleaner, more understandable code. The article demonstrates the value of readily comprehensible software, which in this case improved trust in the science, helped uncover a number of minor bugs, and enabled new kinds of data analysis.

Complementary to the vast enterprise of collecting and processing observational data is the "big science" of climate modeling.[4] Climate scientists use a hierarchy of different models, from simple energy balance models (EBMs), which simply express the basic thermodynamic properties of the planet as a



whole, to general circulation models (GCMs, also known as global climate models) that simulate the movement of mass and energy through the atmosphere via numerical approximations of the equations of fluid motion. Modern climate models usually couple an atmospheric GCM to an ocean GCM, modeling the all-important transfers of energy and moisture between the two major components of the climate system. Increasingly, climate scientists are also coupling GCMs to simulations of other physical and biological processes, such as the carbon cycle, ocean biogeochemistry, atmospheric chemistry, and the dynamics of large ice sheets, to study how these processes interact. With large, highly complex model codes—recalculating the state of the global atmosphere every 10 to 15 simulated minutes—a century-long simulation of global climate might take a month or more to run on a supercomputer, so code optimization and parallelization are important yet often conflict with the scientific need for exact reproducibility.

From a software viewpoint, GCMs typically consist of hundreds of thousands of lines of Fortran that have undergone continual evolution as the science has progressed over the past 40 years. Historically, much of the code was written by the scientists themselves—many with little or no software training—but modeling labs have started hiring software specialists to write, test, and maintain code and to write scripts for configuring and running the models. Today, most major modeling groups have embraced current software techniques such as iterative development, version control, continuous integration, automated testing, and bug tracking.[5] As the complexity of the model code has grown, climate modelers have sought new ways to manage this complexity.

Two of the articles in this issue address such software challenges in GCMs. In "Managing Software Complexity and Variability in Coupled Climate Models," Spencer Rugaber and his colleagues focus on the problem of describing the variability that arises when GCMs are configured to run a huge range of different types of experiments. The article also explores how feature modeling might be used to make sense of this complexity and automatically generate specific model configurations.

Thomas Clune and Richard Rood turn their attention to testing GCMs in "Software Testing and Verification in Climate Model Development." Climate-modeling labs tend to focus their testing efforts on scientific validation of full model runs. Clune and Rood explore how software engineering techniques such as unit tests and test-driven development might be introduced in such an environment, and the challenges of testing numerical methods for which there is no analytical solution.

Although GCMs can help explore the physical science basis for climate change, a different set of models provide support for policymaking (see Figure 1).[6] Integrated assessment models (IAMs) don't simulate the dynamics of atmospheric and ocean processes directly, but instead rely on parameters derived from GCMs (such as climate sensitivity to greenhouse gases, aerosols, and land-use changes), which they combine with models of economic and social change to evaluate proposed climate policy options. Adaptability is important for IAMs, as they need to provide rapid answers during the back and forth of international negotiations. "Enabling Open Development Methodologies in Climate Change Assessment Modeling," by Joshua Introne and his colleagues, addresses a Web services framework for rapidly incorporating and integrating different IAMs.

With a growing need for detailed assessments of the likely impacts of climate change, climate science faces a crisis of scalability—the demands on climate models as inputs to downstream science and policy easily dwarf the number of climate scientists in the world. One possible response might be to nurture broader communities of contributors, much in the way that open source software communities bring together large numbers of dispersed individuals with varying levels of expertise to contribute to a shared goal. Such contributors might help build and test climate software or help run models and conduct experiments as citizen-scientists. A few initiatives, such as climatecode.org and climateprediction.net, have already begun to explore this possibility. The counterargument is that these initiatives don't scale because the processes of developing and validating software, configuring and running simulations, and interpreting results require deep knowledge of the science, and hence will require more intervention from climate experts than can ever possibly be available. This dilemma is the subject of much debate in the climate modeling community, and we're delighted to host this issue's Point/Counterpoint in which Isaac Held of the NOAA Geophysical Fluid Dynamics Laboratory and David Randall of the Atmospheric Sciences Department at Colorado State University present the arguments for and against such an enterprise.

From parallelization to unit testing to designing standardized frameworks for coupling complex multicomponent models, climate science brings crucially important challenges for software specialists. As climate change becomes an increasingly urgent global issue in the coming decades, this challenge will only grow. Along with it will come remarkable opportunities for software professionals working at the interface between science and code.

Acknowledgments

We thank the many anonymous reviewers who provided detailed critical comments and constructive suggestions for improving the articles submitted for this issue.

References

1. K. Richardson et al., "Synthesis Report—Climate Change: Global Risks, Challenges and Decisions," Univ. of Copenhagen, 2009; http://climatecongress.ku.dk/pdf/synthesisreport.
2. P.N. Edwards, *A Vast Machine: Computer Models, Climate Data, and the Politics of Global Warming*, MIT Press, 2010.
3. P. Lynch, "The Origins of Computer Weather Prediction and Climate Modeling," *J. Computational Physics*, vol. 227, no. 7, 2008, pp. 3431–3444; http://linkinghub.elsevier.com/retrieve/pii/S0021999107000952.
4. W.M. Washington, L. Buja, and A. Craig, "The Computational Future for Climate and Earth System Models: On the Path to Petaflop and Beyond," *Philosophical Trans. Series A, Mathematical, Physical, and Eng. Sciences*, vol. 367, no. 1890, 2009, pp. 833–846; www.ncbi.nlm.nih.gov/pubmed/19087933.
5. S.M. Easterbrook and T.C. Johns, "Engineering the Software for Understanding Climate Change," *Computing in Science & Engineering*, vol. 11, no. 6, 2009, pp. 65–74.
6. R.H. Moss et al., "The Next Generation of Scenarios for Climate Change Research and Assessment," *Nature,* vol. 463, no. 7282, 2010, pp. 747–756; www.ncbi.nlm.nih.gov/pubmed/20148028.

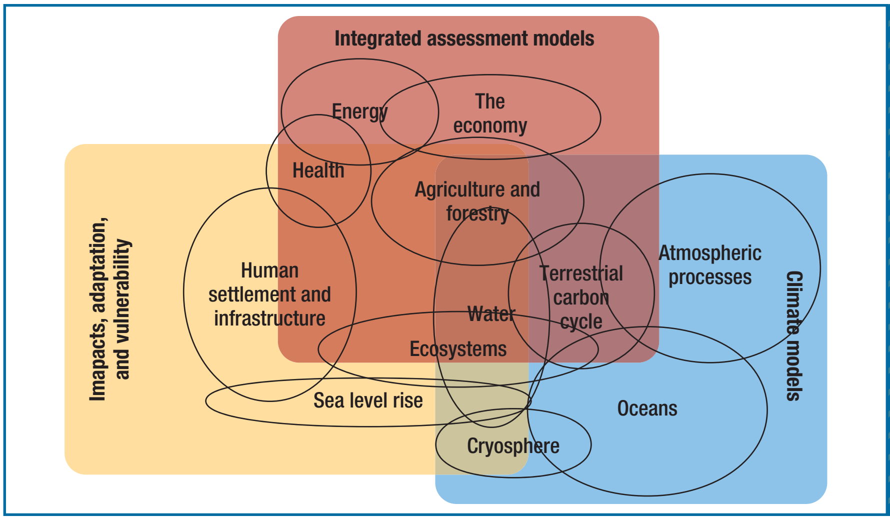


FIGURE 1. Different sets of models (adapted with permission[6]). The models used in climate science and climate policymaking cover different aspects of Earth systems and human activities.

ABOUT THE AUTHORS

STEVE M. EASTERBROOK is a professor of computer science at the University of Toronto. His research interests range from modeling and analysis of complex software systems to the sociocognitive aspects of team interaction, including communication, coordination, and shared understanding in large software teams. Easterbrook received a PhD in computing from Imperial College in London. Contact him at sme@cs.toronto.edu.

PAUL N. EDWARDS is a professor of information and history at the University of Michigan. He initiated a major research project on the history of climate science in 1994, which culminated is his 2010 book *A Vast Machine: Computer Models, Climate Data, and the Politics of Global Warming* (MIT Press). He's also co-editor of *Changing the Atmosphere: Expert Knowledge and Environmental Governance* (MIT Press, 2001), as well as numerous articles. Edwards received a PhD in the history of consciousness program at the University of California, Santa Cruz. Contact him at pne@umich.edu.

VENKATRAMANI BALAJI heads the Modeling Systems Group at the Cooperative Institute of Climate Science, Princeton University. His research interests include parallel computing and scientific infrastructure. Balaji plays advisory roles on various US National Science Foundation, National Oceanic and Atmospheric Administration, and Department of Energy review panels, including a recent series of exascale workshops. He received a PhD in physics and climate science from Ohio State University, and leads workshops on the use of climate models in developing nations such as South Africa and India. Contact him at balaji@princeton.edu.

REINHARD BUDICH is the network manager for the COSMOS Earth system model and IT representative at the Max-Planck-Institute für Meteorologie. He's also technical coordinator of the Infrastructure Project for the European Network for Earth System Modeling. Budich received a diploma in oceanography from Kiel University. Since 2002, he has convened a session called "Earth System Modeling: Strategies and Software" at the annual general assembly of the European Geophysical Union. Contact him at reinhard.budich@zmaw.de.