Clarity in Climate Modeling

Computational models are splendid tools for understanding the intricacies of climate. But can we understand the intricacies of the models?

Brian Hayes

No computer simulations have ever had broader consequences for human life than the current generation of climate models. The models tell us that rising levels of atmospheric carbon dioxide and other greenhouse gases can trigger abrupt shifts in the planet's climate; to avert those changes or mitigate their effects, the entire human population is urged to make fundamental economic and technological adjustments. In particular, we may have to forgo exploiting most of the world's remaining reserves of fossil fuels. It's not every day that the output of a computer program leads to a call for seven billion people to change their behavior.

I hasten to add that computer modeling is not the *only* line of evidence connecting human activities with global warming. We have observations of changes already under way, and there are records of past climate fluctuations showing a close correlation between temperature and atmospheric CO_2. Still, the models provide a crucial link. They offer the only practical way to carry out controlled climate experiments—to change the inputs to the system and see the effect on the outputs. Moreover, the models can reveal causation rather than mere correlation, and they promise insight into the underlying mechanisms of climate change.

As someone interested in both climate and computation—and as a lifelong resident of planet Earth—I have been trying to gain a deeper understanding of how climate models are made. I have been dipping into the primary literature, working through textbooks, sampling the criticisms of global-warming skeptics, browsing the source code of climate models, building a tiny model of my own, and struggling to get a couple of larger models running on my computers. The experience has been rewarding, although the learning curve is steeper than it needs to be. So I have also been thinking about how the basics of climate modeling could be made more widely accessible.

Brian Hayes is senior writer for American Scientist. *Additional material related to the* Computing Science *column can be found online at http://bit-player.org. E-mail: brian@bit-player.org*

Sunshine Equals Earthshine

Underneath all the complexity of a big climate model lies a simple bedrock fact: In the long run, the Earth must balance its energy budget. However much incoming solar radiation the planet absorbs, the same amount must eventually be radiated back into space. The planet warms or cools as needed to satisfy this rule.

A climate model based on energy balance can be simple enough to solve with pencil and paper. It comes down to a single equation:

$$Q(1-\alpha) = \sigma T^4$$

Here sunshine is on the left side of the equal sign, balanced by earthshine on the right. Q is the influx of solar energy averaged over the entire spherical surface of the Earth. The factor α is the planet's albedo, or reflectivity, with a possible range of 0 to 1; thus $1-\alpha$ is the proportion of sunlight absorbed. On the right side of the equation, T is the effective temperature of the Earth, and σ (the Stefan-Boltzmann constant) relates temperature to radiant emission. Because T is raised to the fourth power, even a slight cooling or warming can cause a dramatic dimming or brightening of the planet.

The most interesting thing about this formula is that it seems to give the wrong answer. Plugging in appropriate values of Q, α, and σ, then solving for T, yields a temperature in the neighborhood of –15 degrees Celsius. If the Earth's surface were really that cold, we would be living in an ice age. The true surface temperature, averaged over the entire area of the globe, is about +15 degrees.

This discrepancy was recognized early in the 19th century and resolved in the 1890s by the Swedish scientist Svante Arrhenius, who constructed what deserves to be called the first climate model. The key idea is the greenhouse effect: Incoming sunlight, which is most intense at visible wavelengths, readily passes through the atmosphere, but outgoing emissions at longer, infrared, wavelengths are absorbed by water vapor and carbon dioxide in the air. Seen from afar, the Earth *does* radiate like a body at –15 degrees, but that radiation comes from high in the atmosphere. The surface below is warmed by the blanket of infrared-absorbing gases. It's worth noting that Arrhenius viewed greenhouse warming as a benign effect; only later did accumulating CO_2 begin to look like too much of a good thing.

The Striped Planet

The simplest energy balance calculation reduces the climate of the whole planet to a single number: the average temperature. A little more detail can be coaxed from the model by dividing the globe into latitudinal zones. Each band of latitudes finds its own temperature equilibrium, balancing the inflow and outflow of radiant energy. In addition, energy can be redistributed among the zones as heat flows from warmer to cooler regions.

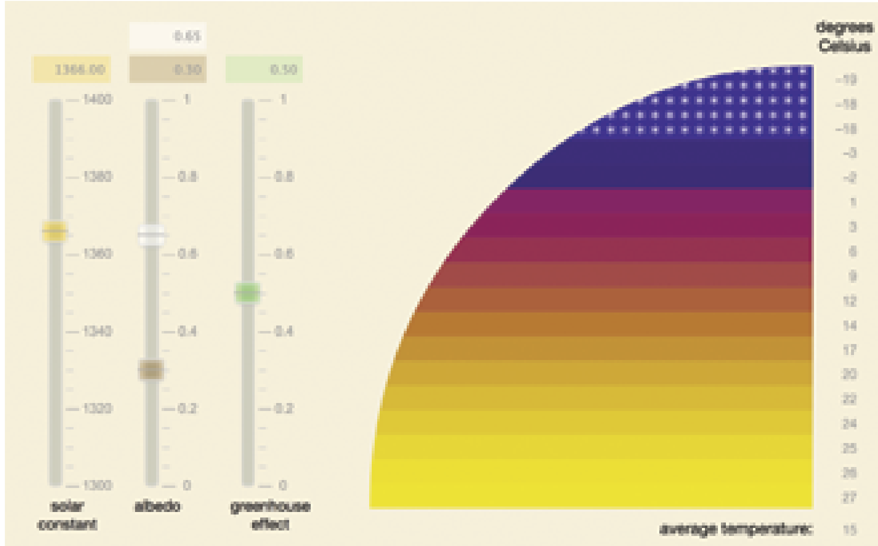
Arrhenius constructed the first such striped model, with latitudinal bands 10 degrees wide. By the 1960s, computers had made it much easier to explore the behavior of the models, testing their response to different inputs or assumptions. Today, you can play with a zonal model yourself, with versions that take the form of BASIC programs, Mathematica notebooks, Java applets, or even Excel spreadsheets. I have written yet another version, which runs in a web browser; see http://bit-player.org/extras/climate.

The zonal models incorporate an important and interesting feedback loop. As already noted, the Earth's temperature depends partly on its albedo, but the albedo also depends partly on the temperature. Any region cold enough to be covered by snow and ice will reflect more light (and absorb less heat) than an uncovered surface. The effect is self-reinforcing: Once an area cools below the freezing point, the diminished absorption of solar energy will cause it to cool still further. The chilled region will also draw heat out of neighboring latitudes, so that they too may freeze. The snow line steadily descends from the poles.

Watching this feedback mechanism at work in the model naturally inspires thoughts of ice age glaciation. Of course the process can also be reversed: Warming at the margins of the temperate zone pushes the snow line poleward, thereby reducing the albedo and bringing further warming. With sufficiently extreme settings in the model, either of these trends can be taken to a limiting state: a "snowball Earth," frozen all the way to the Equator, or a totally ice-free planet.

It's tempting to dismiss these models as cartoonish oversimplifications. Think of all that's been left out: There is no day and night, no summer and winter, no continents and oceans, no winds, clouds, storms, mountains, deserts. On this bald planet, any two places at the same latitude—say Minneapolis and Venice—have the same climate. How ridiculous.

Yet simple models have one winning virtue: We can easily understand what's going on inside them. They shine a spotlight on basic mechanisms and principles. Isaac M. Held of the Geophysical Fluid Dynamics Laboratory in Princeton argues that climate science needs a hierarchy of models, analogous to the hierarchy of experimental organisms in biology, from *E. coli* to *Drosophila* to the mouse. The simpler, smaller models help us understand the big, intricate ones. (Admittedly, even the models at the bottom of Held's proposed hierarchy are more sophisticated than the striped planet described here.)



The simplest climate models balance incoming and outgoing radiant energy. In the model shown here the globe is divided into latitudinal zones, whose mean temperature is encoded in color and indicated numerically in the right margin. The interactive version of the model at http://bit-player.org/extras/climate responds to adjustments in the solar constant, the albedo of land and ice, and a parameter representing the intensity of the greenhouse effect. Such models are primitive, and yet they capture interesting behavior, such as a feedback loop in which cooling promotes ice cover (indicated by snowflakes), which causes further cooling.

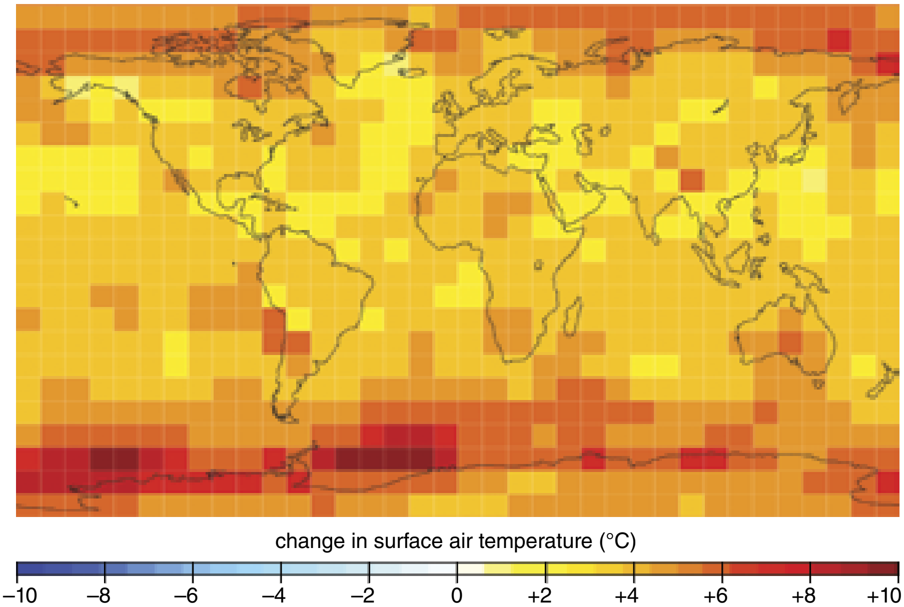
The Swirling Planet

At the other end of the Held hierarchy, the models get very complex indeed. And for good reason: There's a lot going on in the Earth's atmosphere. Water evaporates and condenses; convection cells redistribute both heat and moisture; winds are deflected by the Earth's rotation and break up into turbulent eddies large and small. To represent all this activity, the model atmosphere is sliced up into a three-dimensional lattice with thousands or millions of cells. The model must track flows of energy and mass from cell to cell over time scales ranging from a few seconds to decades.

And it's not enough to model just the atmosphere. Roughly half the heat transported from the tropics to the poles is carried by ocean currents, driven by wind and by gradients in temperature and salinity. The couplings between air and sea produce some of the most distinctive features of the present climatic regime, such as the El Niño Southern Oscillation of the Pacific basin. The most elaborate models incorporate these air–sea interactions as well as the detailed topography of the continents, the dynamics of glaciers and sea ice, the chemical transformations of molecules in the atmosphere, and even biological influences on climate, such as the uptake of carbon dioxide by growing plants.

Mirroring the complexities of the Earth system are the complexities of the software system. A climate model that captures details of all these geophysical processes necessarily becomes a major collaborative programming project. A prominent example is the Community Earth System Model, or CESM, administered by the National Center for Atmospheric Research (NCAR) in Colorado, with contributors at many other institutions. The CESM software has five main modules (for the atmosphere, the oceans, the land, ice on land, and ice on the sea) plus a sixth "coupler" component that coordinates interactions between the subsystems.

Like other climate research groups, the CESM collaboration makes the source code for its models publicly available (see http://www2.cesm.ucar.edu/models). The latest package



The EdGCM model, created by a team at the Goddard Institute for Space Studies, takes about six hours to simulate the evolution of the Earth's climate from 1958 to 2100. In a benchmark measurement of sensitivity to greenhouse gases, the atmospheric CO_2 level is held steady at double the 1958 value. The map plots the resulting change in surface air temperature, comparing average values for 1958–1962 with those projected for 2096–2100. The average temperature change is 3.92 degrees Celsius, but regions near Antarctica warm by almost 10 degrees.

consists of 1.6 million lines of Fortran, with miscellaneous bits and pieces written in more than a dozen other programming languages. The programs and their documentation make fascinating reading, but compiling, configuring, and running them is not a one-click process. I decided that CESM was not the best vehicle for a beginner just learning to drive a climate model.

Models for the Rest of Us

A program called the Educational Global Climate Model (EdGCM) seemed ready-made for my purposes. The software derives from a model developed in the 1980s by James E. Hansen and his colleagues at the Goddard Institute for Space Studies (GISS) in New York. The EdGCM version has been precompiled for easier installation on Windows or Macintosh computers (see http://edgcm.columbia.edu/). It comes packaged with a point-and-click interface as well as tools for plotting and displaying the results of simulations.

In the 1980s Hansen's group used this model in a landmark study of CO_2-induced global warming. The model's atmosphere has nine layers vertically, with horizontal resolution of 8 degrees in latitude by 10 degrees in longitude (a total of roughly 7,000 cells). Atmospheric transport of heat and moisture, as well as the effects of ice, clouds, chemistry, and biology, are all taken into account, although ocean mixing and currents are given a simpler treatment than they are in more recent software.

In the 1990s a summer program for high school and college students offered hands-on experience with data generated by the GISS model, but the students were disappointed that they had no opportunity to run the model itself. To answer this complaint the EdGCM project was launched by Mark Chandler, a member of the GISS climate modeling group. Chandler emphasizes that EdGCM is not a toy but a research tool, functionally identical to the 1980s GISS model. Within Held's hierarchy of climate models, it falls somewhere in the midrange. The source code is only about 20,000 lines, or 1 percent of the CESM total. On the other hand, those 20,000 lines are not easy reading: They were written in a dialect of Fortran that had not yet escaped the age of punch cards and teletype machines.

The program's performance on a modern laptop is far sprightlier than it was on the GISS mainframe 30 years ago. I can calculate about 25 years of simulated climate per hour of running time. Start an experiment at bedtime and it will be finished in the morning.

EdGCM has the potential to open up the world of climate modeling to a much wider public; unfortunately, that potential is going unfulfilled for now. Chandler and his small group struggle to keep the program updated as new computers and mobile devices come along. They also struggle to provide technical support. (I had to ask for help with installation.) Their focus has been on classroom use of the software, and they have not encouraged "hobbyists." Another barrier to casual experimenters is that the software is not free. (The price is $29 for students but up to $199 for others.)

EdGCM is not the only research-level climate model designed for public consumption. Another software package called the Planet Simulator was created by a group at the University of Hamburg Meteorological Institute (see http://www.mi.uni-hamburg.de/plasim). This one is freely distributed. Setting up and running the Hamburg software is somewhat more demanding technically: You'll need to install a Fortran compiler and issue various incantations from the command line. The model has higher spatial resolution than EdGCM, which means it requires a greater commitment of running time and computer resources.

Faith in Numbers

Why would ordinary citizens take an interest in do-it-yourself climate modeling? I can think of two reasons.

First comes the pure pleasure of learning and exploring. Even if climate change had no bearing on human welfare, there would remain a remarkable story of scientific discovery. The mere idea that climate can change is not new, but until recently it was assumed that such changes would proceed very slowly—at a glacial pace. Now we know (mainly from ice cores) that the Earth has a history of quite abrupt and dramatic shifts from one stable state to another, reminiscent of phase transitions. It's like finding out that your friendly, fuddy-duddy insurance agent has led a double life as a tightrope walker. I will never feel quite the same about the planet I live on.

Computational models offer the best prospects for figuring out how these sudden climate shifts come about—what triggers them, and how all the tangled feedback loops interact to amplify small disturbances. And this brings me to the second reason for paying close attention to the models: Global warming *does* have consequences for human welfare, and any successful strategy for addressing the problem will depend on public understanding of what's at stake. Climate scientists have been ringing warning bells for at least 25 years about the effects of atmospheric CO_2; governments and international organizations have responded with various plans and pledges, but carbon emissions continue to grow. The rate of increase in 2013 set a new record.

It's easy to come up with reasons for public inaction: uncertainty about the magnitude of the risk and the cost of the remedies, a reluctance to focus on the future in a world where so many troubles demand immediate attention, the distraction of politicized wrangling between "believers" and "deniers." But the subtleties and complexities of the underlying science are also a factor. It's hard to get motivated about an issue you don't fully understand. Handing out free climate models on street corners may not restore universal reason and civility, but perhaps it's a way of tilting the discussion away from politics and back toward science.

Right now, we have no climate model suitable for handing out on street corners. The ideal model would be effortless to install and run. Or maybe it would run as a web application, which eliminates installation entirely. (Chandler is conducting tests of a web-based follow-on to EdGCM.) The software would have an inviting interface that requires no knowledge of programming, but for those who want to delve deeper it would also be open to exploration of its algorithms and data structures. This last criterion may be the hardest to satisfy. It's not enough to make the source code available; the code must also be written to a high standard of clarity and simplicity.

Should the major climate modeling groups be expected to produce such a pedagogical adjunct to their research tools? In my view it would be entirely appropriate for them to do so. Indeed, because climate change affects every person on Earth, and any remedy will have to enlist the support and cooperation of the whole planet, I find it peculiar that public outreach gets such paltry support. Yet that situation is unlikely to change.

A success story from a few years ago suggests there might be another way. Global warming doubters pressed the Hansen group to publish the source code for a suite of programs known as GISTEMP, which extract temperature trends from historical weather observations. The code was released, but no one could understand it or get it to run. Nick Barnes and David Jones, two British computer scientists, gathered up the untidy collection of files and rewrote them as a single program in the Python language, "with an emphasis on code clarity that encouraged interested people to download, inspect, and run it themselves." Their new program produces the same results as the original. Furthermore, by taking the mystery out of how the temperatures are calculated, they have changed the tenor of the conversation. GISTEMP is no longer so controversial.

Could the same kind of rewriting succeed in the case of a climate model? The undertaking would be substantially more difficult. Modeling programs are larger and more complicated both mathematically and algorithmically. High performance is imperative. Still, it's worth a try. Someday we may be able to say, "Climate modeling—there's an app for that."

For a list of references and links to other resources on climate modeling, see http://bit-player.org/extras/climate.