1

OVERVIEW

MODULE 1.1

Overview of Computational Science

> Scientific revolutions are "those non-cumulative developmental episodes in which in an older paradigm is replaced in whole or in part by an incompatible new one."
>
> —Thomas Kuhn, *The Structure of Scientific Revolutions*

Normally, "the scientific revolution" refers to the discoveries of the sixteenth and seventeenth centuries in Europe, which changed the western view of the natural world. This revolution began with the sun-centered universe of Copernicus and continued until Newton proposed universal gravitation and laws of motion. Nature was the object of much interest, and the exploration of the New World with all its discoveries continued to feed the desire to understand nature.

During the twentieth century, according to the eminent string-theory physicist Michio Kaku, there were three scientific revolutions—the quantum revolution, the biomolecular revolution, and the computer revolution (Kaku 1998). Few can doubt the rapidity with which recent scientific advances have been made with each new discovery or insight changing our view of our planet, its inhabitants, and often the universe. The accomplishments of that century augurs very well for the current one.

Early in the twenty-first century, Microsoft Research convened a workshop of international authorities to devise a "vision and roadmap of the evolution, challenges and potential of computer science and computing in scientific research during the next fifteen years." The outcome was "Towards 2020 Science." What they predicted marks the beginnings of a new scientific revolution, where computation will become more than an adjunct supporter of scientific research. Computational principles and tools will become integrated into science, changing the fundamental way that science is practiced. Computational science in both theoretical and experimental sciences will greatly augment the rates of scientific advances that will benefit the planet and our species (Microsoft Research 2006). For example, the results of the human genome project, which depended upon large-scale computational science, have encouraged a myriad of new research and development in government, university, and commercial laboratories. One significant outcome from these projects will be a far better understanding of molecular mechanisms that underlie human diseases and their more effective treatments.

In 2005, the President's Information Technology Advisory Committee released the report "Computational Science: Ensuring America's Competiveness" (Report to the President 2005). They concluded that computational science and high-performance computing could be integral to innovations in all of the sciences (biological/biomedical, physical, and social), engineering, industry, and defense. Advances in computation allow us to acquire and analyze enormous streams of data, making it possible to consider and solve problems heretofore unapproachable. Computational science also allows us to build models, visualize phenomena, and conduct experiments difficult or impossible in the laboratory. We can now examine interactions in systems that involve more than one discipline, encouraging us to collaborate with specialists in other fields. Such collaboration should lead to solutions that are creative, synergistic, sustainable, and economically favorable.

Computational science, the fast-growing interdisciplinary field that is at the intersection of the sciences, computer science, and mathematics, will require scientists who are appropriately trained. The experts who produced "Towards 2020 Science" predicted that future scientists who are not computationally and mathematically literate will be unable to do science. Chemistry professor Robert Harrison, director of the Joint Institute for Computational Sciences at the University of Tennessee, states in the JICS Mission webpage, "To translate even the most elementary theories into useful tools for physical chemistry discovery, you have to do large-scale computation." He states further, "If you look at students coming into our graduate program from the undergraduate world, those that haven't already had some exposure to computation, such as thinking algorithmically, solving problems on the computer, and the little bits of applied math that you need to understand all of that, . . . have lost a year or two of productivity at the graduate level. But it's not only the undergraduate students coming into graduate school that have this issue; it's also our undergrads going off into the larger world. Industry and many other aspects of the commercial world use simulation and computation in diverse ways" (JICS).

Computational science, which combines computer simulation, scientific visualization, mathematical modeling, computer programming, data structures, networking, database design, symbolic computation, and high-performance computing, can transform practices in a diverse range of disciplines. Its computer models and simulations offer valuable approaches to problems in many areas, as the following examples indicate.

1. Scientists at Los Alamos National Laboratory and the University of Minnesota wrote, "Mathematical modeling has impacted our understanding of HIV pathogenesis. Before modeling was brought to bear in a serious manner, AIDS was thought to be a slow disease in which treatment could be delayed until symptoms appeared, and patients were not monitored very aggressively. In the large, multicenter AIDS cohort studies aimed at monitoring the natural history of the disease, blood typically was drawn every six months. There was a poor understanding of the biological processes that were responsible for the observed levels of virus in the blood and the rapidity at which the virus became drug resistant. Modeling, coupled with advances in technology, has changed all of this." Dynamic modeling has not only revealed important features of HIV pathogenesis but has advanced the drug treatment regime for AIDS patients (Perelson and Nelson 1999). Since then, Perelson and other researchers have applied modeling to enhance our understanding of the hepatitis C virus, which causes widespread infections and is the primary cause of liver cancer in the United States. Such models have already revealed much about the pathogenesis of the virus, the effectiveness of treatments (interferon/ribavirin and direct antiviral agents), and the influence of genetic variants in the kinetics of the virus (Dahari et al. 2011).

2. From the 1960s, numerical weather prediction has revolutionized forecasting. "Since then, forecasting has improved side by side with the evolution of computing technology, and advances in computing continue to drive better forecasting as weather researchers develop improved numerical models" (Pittsburgh Supercomputing Center 2001). A Weather Research and Forecasting (WRF) Model was released in 2000. The latest version of this model utilizes "multiple dynamical cores, a 3-dimensional variational data assimilation system, and a software architecture allowing for computational parallelism and system extensibility." This sophisticated, mesoscale [horizontal scale of 2 to 2000 kilometers (km)] numerical weather-prediction system is useful for forecasting and research. An array of partners, including the National Center for Atmospheric Research (NCAR), the National Oceanic and Atmospheric Administration (NOAA), other governmental and military organizations, universities, plus some international groups, continuously revise the WRF model. With this efficient, adaptable model for forecasting, researchers can conduct simulations using real data or idealized designs. In 2007, a specialized version of WRF was initiated for forecasting and research on hurricanes (WRF).

3. A multidisciplinary team at the University of Tennessee's Institute for Environmental Modeling is using computational ecology to study complex options for ecological management of the Everglades. Louis Gross, Director of the Institute, says that "computational technology, coupled with mathematics and ecology, will play an ever-increasing role in generating vital information society needs to make tough decisions about its surroundings" (Lymn 2003). South Florida has a well-known history of disruptions to normal water flow. The UT group has developed a parallelized landscape population model (ALFISH) to integrate with other models in a multiscale ecological multimodel (Across-Trophic Level System Simulation, ATLSS). ALFISH is used to model the effects on freshwater fish (planktivorous and piscivirous) of different water-management plans. These fish populations represent food resources for wading birds, and researchers can link the fish model with wading bird models to help sustain the higher-level multimodel (Wang et al. 2006).

4. Application of computer modeling has fueled the debate in another, rather unexpected area—linguistics. The origin of the Indo-European family of languages is rather hotly debated between proponents of two hypotheses—Eurasian steppes, 6000 years (yr) ago versus Anatolia (mostly in present-day Turkey), 8000 to 9500 yr ago. This family of languages has given rise to more than 400 modern languages, spoken by about three billion people. Recently, Quentin Atkinson and colleagues utilized evolutionary models, often employed to ascertain the origin of viruses that lead to epidemics, to analyze this problem. Based on common vocabulary words from various languages in the family, the model supports Anatolian origin, as agricultural techniques were broadcast. Although certainly not resolving the argument, the results have given experts in this field something to consider carefully. (Bouckaert et al. 2012)

Projects

1. Investigate three applications of computational science involving different scientific areas and write at least a paragraph on each. List references.
2. Investigate an application of computational science and write a three-page, typed, double-spaced paper on the topic. List references.

References

Bouckaert, Remco, Philippe Lemey, Michael Dunn, Simon J. Greenhill, Alexander V. Alekseyenko, Alexei J. Drummond, Russell D. Gray, Marc A. Suchard, and Quentin D. Atkinson. 2012. "Mapping the Origins and Expansion of the Indo-European Language Family." *Science* 337(6097): 957–960.

Dahari, H., J. Guedj, A. Perelson, and T. Layden. 2011. "Hepatitis C Viral Kinetics in the Era of Direct Acting Antiviral Agents and IL28B." *Curr Hepat Rep.* 10(3): 214–227. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3237049/pdf/nihms341709.pdf (accessed August 12, 2012)

JICS (Joint Institute for Computational Science). University of Tennessee and Oak Ridge National Laboratory. "Harrison Discusses JICS Mission." http://www.jics.tennessee.edu/jics-mission (accessed August 20, 2012)

Kaku, Michio. 1998. *Visions: How Science Will Revolutionize the 21st Century*. New York: Anchor.

Kuhn, Thomas. 1962. *The Structure of Scientific Revolutions.* Chicago: University of Chicago Press.

Lymn, Nadine. 2003. "Computational Ecology." Math & Bio 2010: Linking Undergraduate Disciplines, Kickoff Meeting—Articles and Reports. http://www.maa.org/mtc/Computational-Ecology.pdf (accessed January 7, 2013)

Microsoft Research. 2006. "Towards 2020 Science." http://research.microsoft.com/en-us/um/cambridge/projects/towards2020science/ (accessed August 12, 2012)

Perelson, Alan S., and Patrick W. Nelson. 1999. "Mathematical Analysis of HIV-1 Dynamics in Vivo." *SIAM REVIEW*, 41(1): 3–44.

Pittsburgh Supercomputing Center. 2001. "Pittsburgh System Marks a Watershed in Weather Prediction." News Release, December 4, 2001. http://www.psc.edu/publicinfo/news/2001/weather-12-04-01.html (accessed January 7, 2013)

Report to the President. June 2005. "Computational Science: Ensuring America's Competitiveness." President's Information Technology Advisory Committee. National Coordination Office for Information Technology Research and Development (NCO/IT R&D). http://www.nitrd.gov/pitac/reports/20050609_computational/computational.pdf (accessed January 7, 2013)

Wang, D., M. W. Berry, E. A. Carr, and L. J. Gross. 2006. "A Parallel Fish Landscape Model for Ecosystem Modeling." *SIMULATION* 2006, 82: 451.

WRF (The Weather Research and Forecasting Model). http://wrf-model.org/index.php (accessed August 23, 2012)

MODULE 1.2

The Modeling Process

Introduction

The process of making and testing hypotheses about models and then revising designs or theories has its foundation in the experimental sciences. Similarly, computational scientists use **modeling** to analyze complex, real-world problems in order to predict what might happen with some course of action. For instance, Professor Muneo Hori and colleagues from the Earthquake Research Institute, the University of Tokyo, Japan, use high-performance computation with sophisticated models to simulate earthquakes, making quantitative predictions of infrastructural damages, response, and recovery to help minimize damage, death, and injury (Lalith and Hori 2012). Professor Liming Liang, statistical geneticist at Harvard School of Public Health, uses computational and statistical tools to better understand the genetic variation in complex human diseases, such as dyslipidemia, cancer, and type 2 diabetes (Liang). Cognitive scientists, such as Professor Ken Koedinger at the Human-Computer Interaction Institute, Carnegie-Mellon University, develop computer models of student reasoning and learning to aid in the design educational software and to guide teaching practices (Koedinger). Both civilian and military organizations commonly employ drones (unmanned aerial vehicles, UAVs). Whether to monitor air quality or supervise combat forces, this technology is becoming more and more important, but the operation of the drones is quite complex. While a postdoctoral fellow at MIT, Dr. Luca Bertuccelli worked with a team using models to develop new decision support systems, enabling the operators of these craft to make better decisions (Bertuccelli). Arboviruses are arthropod-borne viruses that cause diseases, such as West Nile encephalitis, dengue fever, and yellow fever. A mathematical modeling team at the Universidad Nacional Autónoma de México has been modeling the dynamics of such infections. A better understanding of these viruses will improve outbreak predictions, interventions, and responses (Vargus and Cruz-Pacheco 2010).

Definition **Modeling** is the application of methods to analyze complex, real-world problems in order to make predictions about what might happen with various actions.

Model Classifications

Several classification categories for models exist. A system we are modeling exhibits **probabilistic**, or **stochastic**, **behavior** if it appears that an element of chance exists. For example, the path of a hurricane is probabilistic. In contrast, a behavior can be **deterministic**, such as the position of a falling object in a vacuum. Similarly, models can be deterministic or probabilistic. A **probabilistic**, or **stochastic**, **model** exhibits random effects, while a **deterministic model** does not. The results of a deterministic model depend on the initial conditions; and in the case of computer implementation with particular input, the output is the same for each program execution. As we see in Module 9.2, "Simulations," and other modules, we can have a probabilistic model for a deterministic situation, such as a model that uses random numbers to estimate the area under a curve.

Definitions A system exhibits **probabilistic**, or **stochastic**, **behavior** if an element of chance exists. Otherwise, the system exhibits **deterministic behavior**. A **probabilistic**, or **stochastic**, **model** exhibits random effects, while a **deterministic model** does not.

We can also classify models as static or dynamic. In a **static model**, we do not consider time, so that the model is comparable to a snapshot or a map. For example, a model of the weight of a salamander as being proportional to the cube of its length has variables for weight and length but not for time. By contrast, in a **dynamic model**, time changes, so that such a model is comparable to an animated cartoon or a movie. For example, the number of salamanders in an area undergoing development changes with time; hence, a model of such a population is dynamic. Many of the models we consider in this text are dynamic and employ a static component as part of the dynamic model.

Definitions A **static model** does not consider time, while a **dynamic model** changes with time.

When time changes continuously and smoothly, the model is **continuous**. If time changes in incremental steps, the model is **discrete**. A discrete model is analogous to a movie. A sequence of frames moves so quickly that the viewer perceives motion. However, in a live play, the action is continuous. Just as a discrete sequence of movie frames represents the continuous motion of actors, we often develop discrete computer models of continuous situations.

Definitions In a **continuous model**, time changes continuously, while in a **discrete model**, time changes in incremental steps.

Steps of the Modeling Process

The modeling process is cyclic and closely parallels the scientific method and the software life cycle for the development of a major software project. The process is cyclic because at any step we might return to an earlier stage to make revisions and continue the process from that point.

The steps of the modeling process are as follows:

1. **Analyze the problem.**
   We must first study the situation sufficiently to identify the problem precisely and understand its fundamental questions clearly. At this stage, we determine the problem's objective and decide on the problem's classification, such as deterministic or stochastic. Only with a clear, precise problem identification can we translate the problem into mathematical symbols and develop and solve the model.
2. **Formulate a model.**
   In this stage, we design the model, forming an abstraction of the system we are modeling. Some of the tasks of this step are as follows:
   a. **Gather data.**
      We collect relevant data to gain information about the system's behavior.
   b. **Make simplifying assumptions and document them.**
      In formulating a model, we should attempt to be as simple as reasonably possible. Thus, we frequently decide to simplify some of the factors and to ignore other factors that do not seem as important. Most problems are entirely too complex to consider every detail, and doing so would only make the model impossible to solve or to run in a reasonable amount of time on a computer. Moreover, factors often exist that do not appreciably affect outcomes. Besides simplifying factors, we may decide to return to Step 1 to restrict further the problem under investigation.
   c. **Determine variables and units.**
      We must determine and name the variables. An **independent variable** is the variable on which others depend. In many applications, time is an independent variable. The model will try to explain the **dependent variables**. For example, in simulating the trajectory of a ball, time is an independent variable; and the height and the horizontal distance from the initial position are dependent variables whose values depend on the time. To simplify the model, we may decide to neglect some variables (such as air resistance), treat certain variables as constants, or aggregate several variables into one. While deciding on the variables, we must also establish their units, such as days as the unit for time.
   d. **Establish relationships among variables and submodels.**
      If possible, we should draw a diagram of the model, breaking it into submodels and indicating relationships among variables. To simplify the model, we may assume that some of the relationships are simpler than they really are. For example, we might assume that two variables are related in a linear manner instead of in a more complex way.
   e. **Determine equations and functions.**

While establishing relationships between variables, we determine equations and functions for these variables. For example, we might decide that two variables are proportional to each other, or we might establish that a known scientific formula or equation applies to the model. Many computational science models involve differential equations, or equations involving a derivative.

3. **Solve the model.**
   This stage implements the model. It is important not to jump to this step before thoroughly understanding the problem and designing the model. Otherwise, we might waste much time, which can be most frustrating. Some of the techniques and tools that the solution might employ are algebra, calculus, graphs, computer programs, and computer packages. Our solution might produce an exact answer or might simulate the situation. If the model is too complex to solve, we must return to Step 2 to make additional simplifying assumptions or to Step 1 to reformulate the problem.
4. **Verify and interpret the model's solution.**
   Once we have a solution, we should carefully examine the results to make sure that they make sense (verification) and that the solution solves the original problem (validation) and is usable. The process of **verification** determines if the solution works correctly, while the process of **validation** establishes whether the system satisfies the problem's requirements. Thus, verification concerns "solving the problem right," and validation concerns "solving the right problem." Testing the solution to see if predictions agree with real data is important for verification. We must be careful to apply our model only in the appropriate ranges for the independent data. For example, our model might be accurate for time periods of a few days but grossly inaccurate when applied to time periods of several years. We should analyze the model's solution to determine its implications. If the model solution shows weaknesses, we should return to Step 1 or 2 to determine if it is feasible to refine the model. If so, we cycle back through the process. Hence, the cyclic modeling process is a trade-off between **simplification** and **refinement**. For refinement, we may need to extend the scope of the problem in Step 1. In Step 2, while refining, we often need to reconsider our simplifying assumptions, include more variables, assume more complex relationships among the variables and submodels, and use more sophisticated techniques.
5. **Report on the model.**
   Reporting on a model is important for its utility. Perhaps the scientific report will be written for colleagues at a laboratory or will be presented at a scientific conference. A report contains the following components, which parallel the steps of the modeling process:
   a. **Analysis of the problem**
      Usually, assuming that the audience is intelligent but not aware of the situation, we need to describe the circumstances in which the problem arises. Then, we must clearly explain the problem and the objectives of the study.
   b. **Model design**
      The amount of detail with which we explain the model depends on the situation. In a comprehensive technical report, we can incorporate

much more detail than in a conference talk. In either case, we should state the simplifying assumptions and the rationale for employing them. Clearly labeled diagrams of the relationships among variables and submodels are usually very helpful in understanding the model.

c. **Model solution**
In this section, we describe the techniques for solving the problem and the solution. We should give as much detail as necessary for the audience to understand the material without becoming mired in technical minutia. For a written report, appendices may contain more detail, such as source code of programs and additional information about the solutions of equations.

d. **Results and conclusions**
Our report should include results, interpretations, implications, recommendations, and conclusions of the model's solution. Usually, we present some of the data and results in tables or graphs. Such figures should contain titles, sources, and labels for columns and axes. We may also include suggestions for future work.

6. **Maintain the model.**
As the model's solution is used, it may be necessary or desirable to make corrections, improvements, or enhancements. In this case, the modeler again cycles through the modeling process to develop a revised solution.

Definitions The process of **verification** determines if the solution works correctly, while the process of **validation** establishes if the system satisfies the problem's requirements.

Although we described the modeling process as a sequence or series of steps, we may be developing two or more steps simultaneously. For example, it is advisable to be compiling the report from the beginning. Otherwise, we can forget to mention significant points, such as reasons for making certain simplifying assumptions or for needing particular refinements. Moreover, within modeling teams, individuals or groups frequently work on different submodels simultaneously. Having completed a submodule, a team member might be verifying the submodule while others are still working on solving theirs.

The modeling process is a creative, scientific endeavor. As such, a problem we are modeling usually does not have one correct answer. The problems are complex, and many models provide good, although different, solutions. Thus, modeling is a challenging, open-ended, and exciting venture.

Exercises

1. Compare and contrast the modeling process with the scientific method: Make observations; formulate a hypothesis; develop a testing method for the hypothesis; collect data for the test; using the data, test the hypothesis; accept or reject the hypothesis.

2. Compare and contrast the modeling process with the software life cycle: analysis, design, implementation, testing, documentation, maintenance.

References

Bertuccelli, Luca F. Personal webpage. http://web.mit.edu/lucab/www/publications.html (accessed August 29, 2012)

Giordano, Frank R., Maurice D. Weir, and William P. Fox. 2003. *A First Course in Mathematical Modeling*. 3rd ed. Pacific Grove, CA: Brooks/Cole-Thompson Learning.

Koedinger. K. Personal webpage. http://www.hcii.cmu.edu/people/faculty/ken-koedinger (accessed August 28, 2012)

Lalith, Maddegedara and Muneo Hori. 2012. "Application of High Performance Computation for the Prediction of Urban Area Earthquake Disaster." Presented to the 8th International Symposium on Social Management Systems SSMS2012 (May), Kaohsiung, Taiwan. http://management.kochi-tech.ac.jp/ssms_papers/sms11-4706_4a089987c005e671dd603ce4162e6704.pdf (accessed August 28, 2012)

Liang, Liming. My Research, personal webpage. http://www.hsph.harvard.edu/faculty/liming-liang/index.html (accessed August 28, 2012)

Vargus, C., L. Esteva, and G. Cruz-Pacheco. 2010. "Mathematical Modelling of Arbovirus Diseases." Electrical Engineering Computing Science and Automatic Control (CCE), 2010 7th International Conference (September 8–10, 2010, Mexico City). pp. 205–209.