

An Introduction to Agent-Based Modeling for Undergraduates

Angela B. Shiflet¹ and George W. Shiflet¹

¹Wofford College, Spartanburg, SC USA

shifletab@wofford.edu, shifletgw@wofford.edu

Abstract

Agent-based modeling (ABM) has become an increasingly important tool in computational science. Thus, in the final week of the 2013 fall semester, Wofford College's undergraduate Modeling and Simulation for the Sciences course (COSC/MATH 201) considered ABM using the NetLogo tool. The students explored existing ABMs and completed two tutorials that developed models on unconstrained growth and the average distance covered by a random walker. The models demonstrated some of the utility of ABM and helped illustrate the similarities and differences between agent-based modeling and previously discussed techniques—system dynamics modeling, empirical modeling, and cellular automaton simulations. Improved test scores and questionnaire results support the success of the goals for the week.

1 Introduction

1.1 Agent-Based Modeling in Economics and Science

On September 18, 2008, the Wall Street Journal ran the headline—“Mounting Fears Shake World Markets As Banking Giants Rush to Raise Capital.” The New York Times headline on September 19, 2008 blared, “Vast Bailout by U.S. Proposed in Bid to Stem Economic Crisis.” Alan Greenspan, then chairman of the Federal Reserve Board, described the financial crisis as “much broader than anything [he] could have imagined.” He stated that he had “found a flaw in the model that [he] perceived [to be] the critical functioning structure that defines how the world works” (Hearing 2008). Stirred by the economic crisis and confusion, the Subcommittee on Science and Technology from the U.S. House of Representatives held hearings in 2010, focusing on “promise and limits of modern macroeconomic theory in light of the current economic crisis” (Building 2010).

Then and now, the predominating macroeconomic model of the economy was and is the Dynamic Stochastic General Equilibrium (DSGE) model. This model supposes that all prices or values approach some ideal equilibrium and that markets reflect a balance of forces, such as supply and

demand. Prices and markets are assumed to lack any intrinsic, internal dynamics, only reacting to new input. The model excludes factors, such as subprime mortgages and financial derivatives, which might introduce “nonlinearity” to the mix. Unfortunately, this equilibrium-type model is the one routinely consulted by the Federal Reserve and other central banks. Economists are a cautious lot and are reluctant to consider other types of models, which possibly simulate better a very complex economic system. Under normal, calm economic conditions, the DSGE model might work very well, but such conditions do not always exist. Economic crises are not considered.

While everyone scurried about trying to figure out “why the sky was falling” and who was responsible, some academics proposed an explanation. In a 2008 Op-Ed in the *New York Times*, theoretical physicist Mark Buchanan wrote that financial crises might derive naturally from the market structure and that some economists suggested a new type of model might better simulate a real economy. These models are termed agent-based (Buchanan 2008).

Agent-based models (ABMs) simulate the interactions between agents. If applied to the economy, the agents might be individual consumers, government institutions, financial institutions, or policy makers. In the model, a set of rules govern each type of agent’s behavior. Agents act at any point in time according to their own situation, the environment, and their rules of behavior. For instance, a consumer might decide to buy or not to buy a new car for many reasons. The decision might be based on need, optimism about the economy or inflation, current employment, financial situation, and/or the perception of a “good deal.” The actions are not grounded in some overarching assumption of equilibrium. The agents interact directly with other agents, and this interaction will likely change their conduct. Such behaviors more accurately reflect those occurring in the real economy, so that overall market behavior emerges from the actions of all these interacting agents (Chan, Son, and Macal 2010).

ABMs have revealed that instability in the markets does not develop gradually but appears very quickly. Upon achieving a threshold, a swift collapse occurs. For instance in the U.S. housing market, overall debt accumulated from increasing home prices, lower interest rates, and easy credit (Economist 2010). With increasing amounts of debt, agents of the economy become connected more tightly in a complex state of interdependence. If problems befall one agent, the difficulties are likely to spread throughout the system. With access to this type of model, policy makers could experiment with the effects of different government actions and policies that might avert these types of calamitous events, or at least minimize their consequences (Buchanan 2008).

Thus, it appears that economists should follow the lead of the physical, biological, and other social sciences in using the currently available computational power to develop models that more appropriately deal with the complexity of the economy and its emergent characteristics. For instance, biologists and computational scientists at the University of Sheffield have developed the Epitheliome Project, where they use agent-based models that “predict the social behavior of cells in epithelial tissue” (Epitheliome Project). At the Center for Advanced Modeling at Johns Hopkins School of Medicine, scientists developed distributed platform, agent-based models for disease transmission simulating disease outbreaks in a population of billions agents (Parker and Epstein 2001). Also, Dr. Lee Hoffer, medical anthropologist at Case Western Reserve University, is using ABMs to better understand heroin markets (Hoffer 2013). ABM simulations can empower and inform economists and policy makers as these models are informing many scientists. Such models reveal relationships that are not perceptible to even the brightest of human minds.

1.2 Agent-Based Modeling in the Classroom

Because of its utility, agent-based modeling was introduced to undergraduates in two sections (31 students, in all) of Wofford College's Modeling and Simulation for the Sciences course (COSC/MATH 201) in fall, 2013. COSC/MATH 201, with a prerequisite of one semester of calculus, is dual-listed as a second-year computer science and mathematics course. Wofford's Emphasis in

Computational Science (ECS 2014), a program for bachelor-of-science students that is similar to a minor, requires COSC/MATH 201, while the course can count for majors in computer science, environmental studies, and mathematics and minors in computer science and mathematics.

Previously, the class had covered simulation methods and error (about 1.5 weeks) along with three other modeling techniques—system dynamics modeling (about 4.5 weeks), empirical modeling (about 1.5 weeks), and cellular automaton simulations (about 4.5 weeks). During a one-week period at the end of the semester, students worked through two tutorials that helped develop their understanding of agent-based modeling as it compares to these other methods.

1.3 Agent-Based Modeling in the Textbook and Online

COSC/MATH 201 used as a textbook the first edition of *Introduction to Computational Science: Modeling and Simulation for the Sciences*, which the authors of this paper wrote (Shiflet and Shiflet 2008, 2014). The second edition contains a new chapter on "Agent-Based Models." The associated website (Introduction to Computational Science 2014) contains the two tutorials discussed in this paper for the NetLogo[®] package, developed by the first author and Wofford student Richard Mugabe, as well as AgentSheets versions of these tutorials and additional tutorials for both tools. Answers to the tutorials are available to instructors who adopt the text.

2 Modeling Techniques in the Course

2.1 System Dynamics Modeling

Modeling and Simulation for the Sciences began with a study of system dynamics models, which consider the changing sizes of complex interrelated systems as time progresses. For example, such a model of spread of flu might display the number of susceptibles, infecteds, and recovered at each time step. The system dynamics tool STELLA[®] lowered barriers for developing such models by facilitating visualization of the relationships among systems, entry of appropriate equations, generation of simulations, and creation of graphs and tables of the results. Starting with a system dynamics model involving unconstrained growth, the class progressed to models on constrained growth, drug dosage, competition, predator-prey, and spread of disease. Two tutorials, developed for the textbook, covered the requisite information for the subsequent material in a just-in-time fashion. Using STELLA, the students in pairs developed system dynamics models for a variety of applications, including diffusion of heat, radioactive decay chains, scuba diving, the carbon cycle, antibiotic resistance, and the economics of fishing.

2.2 Empirical Modeling

Sometimes, however, it is difficult or impossible to develop a mathematical model that explains a situation. Thus, after examining system dynamics modeling, several numerical integration techniques, and computational error and working through just-in-time MATLAB tutorials for the course, the class spent several days on empirical modeling. Learning that an empirical model is based only on data and is used to predict, not explain, a system, the students were introduced to techniques for finding functions that fit the data. Thus, although an empirical model could not explain a system, students could use such a model to predict behavior where data do not exist, to suggest a model, to estimate its parameters, and to test a model.

2.3 Cellular Automaton Simulations

Then, for about one month, the students considered the important technique of cellular automaton simulations, developed with MATLAB®. Cellular automata are dynamic computational models that are discrete in space, state, and time. We picture space as a one-, two-, or three-dimensional grid, or array. A site, or cell, of the grid has a state, and the number of states is finite. Transition rules, specifying local relationships and indicating how cells are to change state, regulate the behavior of the system. An advantage of such grid-based models is that we can visualize the progress of events through informative animations. For example, we can view a simulation of heat diffusing through a metal bar or the movement of ants toward a food source.

Besides covering models for random walk, the diffusion of heat, and the spreading of fire, pairs of students had two major projects. The first involved simulating the growth of polymers or solidification. The second simulation involved applications in foraging, hunting by a pit viper, growth of mushroom fairy rings, HIV developing in the body, or invasion by exotic plants.

2.4 Agent-Based Modeling

The last week of the class consisted of students working through two tutorials on agent-based models using NetLogo. The students learned that with an agent-based model (ABM), each entity, such as an animal, is modeled as an autonomous, decision-making agent that has a state, which is represented by a set of state variables, and behaviors, which control its actions. A method or procedure, which is associated with a group of agents, captures some or all of an agent's behavior. Agents often operate in an environment that is a rectangular grid of cells. The environment, its neighboring agents, and the states and behavior of an agent determine the agent's new state. For each time step, instead of iterating through each grid cell as with a cellular automaton simulation, an agent-based simulation proceeds through each agent, revising its state.

3 Agent-Based Tutorials

Monday of the last week of classes for Modeling and Simulation for the Sciences began with a pretest and a brief introduction to agent-based modeling and its utility in science. Moreover, ABMs of similar applications to those covered earlier in the semester, such as predator-prey, spreading of disease, and progression of fire, were demonstrated, compared, and contrasted. The students then began working through the first tutorial in the 50-minute class with assistance from the instructor. Completing the tutorial outside of class, perhaps with their own copy of NetLogo, which is free to download, they submitted the NetLogo file and answers to the tutorial's quick review questions before Wednesday's class. During this subsequent class and afterwards, the students completed a second tutorial, which they submitted before Friday's class.

3.1 ABM Tutorial 1

Emphasizing the utility of ABMs, in a step-by-step fashion the first tutorial, with a series of quick review questions, leads the students through employing NetLogo to develop an agent-based simulation of unconstrained growth of bacteria. Figure 1 displays one version of the resulting code a student might produce, while Figure 2 shows the result of a simulation. As visualized, in contrast to an earlier system dynamics model (Figure 3), which only shows the number of bacteria changing with time (Figure 4), this ABM displays local views of individuals as well as an associated graph of the unconstrained bacteria's exponential growth. The NetLogo output is visually appealing and overcomes the abstraction of a strictly graphical representation. Observing that each bacterium can divide with

probability *growth-rate* at each time step, agent-based simulation with visualization can enhance the student's understanding of exponential growth in a variety of applications, from growth of bacteria to growth of a bank account.

```

; Unconstrained growth simulation

globals [
  growth-rate
  chance-move
]

; put red turtle at location (0, 0)
to setup
  clear-all
  set-default-shape turtles "circle"
  crt 1 [ set color red ]
  set growth-rate 0.1
  set chance-move 0.15
  reset-ticks
end

; master scheduler
; A bacterium moves with probability chance-move.
; A bacterium divides with a growth rate of growth-rate.
; Simulation stops after 100 ticks.
to go
  ask turtles [
    if random-float 1.0 < chance-move [
      move-to one-of neighbors
    ]
  ]

  ask turtles [
    if random-float 1.0 < growth-rate [
      hatch 1
    ]
  ]

  tick
  if ticks >= 100 [stop]
end

```

Figure 1: NetLogo code for unconstrained growth model

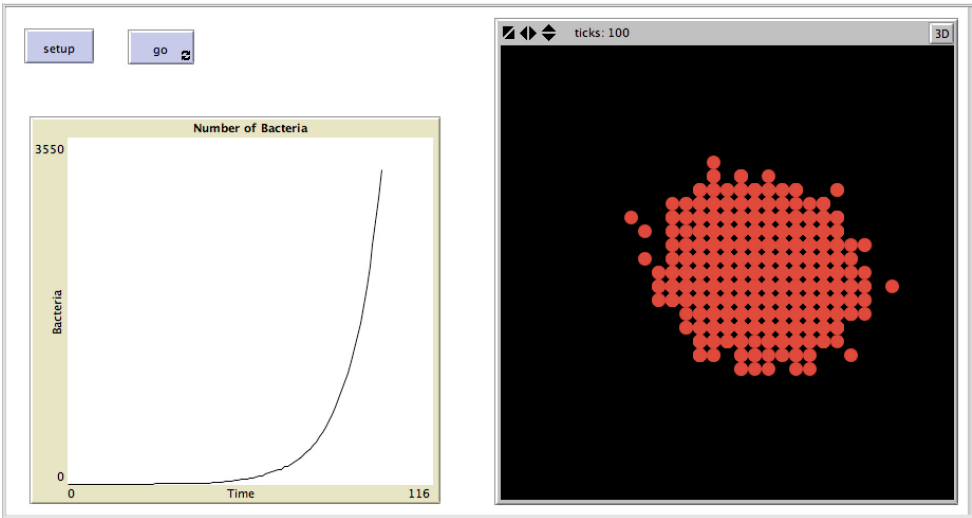


Figure 2: Interface level after one simulation of unconstrained growth code in Figure 1

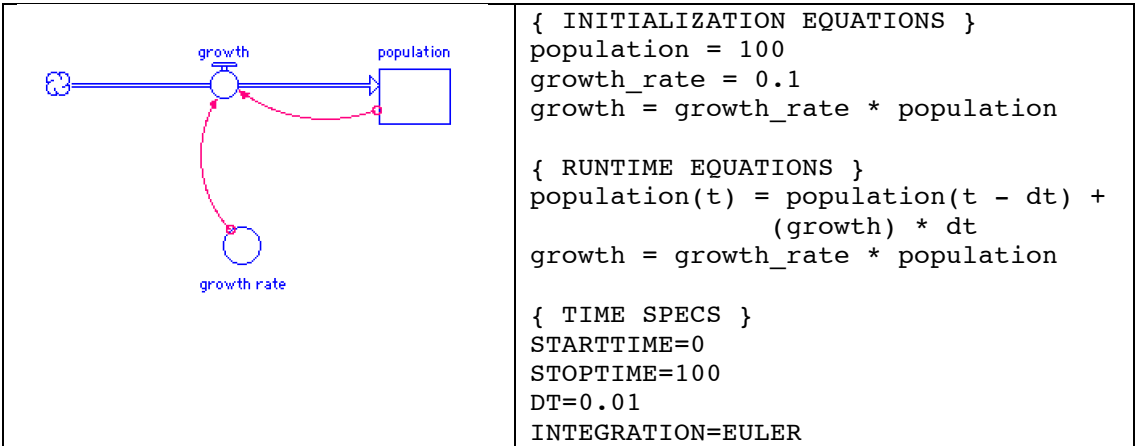


Figure 3: STELLA diagram and difference equations for unconstrained growth model

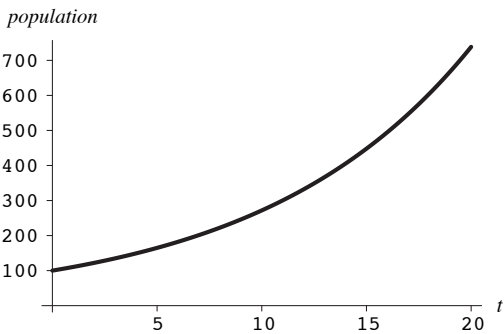


Figure 4: Output from model in Figure 3

3.2 ABM Tutorial 2

As with unconstrained growth, students had studied random walks earlier in the semester in a different context. Individuals had developed in class, with the help of the instructor, a random-walk simulation and animation in MATLAB. Building on this background, each student wrote a MATLAB function, *howFar*, with parameter n that returned the distance traveled from the origin (the start) in n steps. Finally, for homework, students developed a function, *avgHowFar*, with parameters n and *numTimes* that called *howFar*(n) for *numTimes* number of times and returned the average distance traveled. As indicated in the text, a plot of the average distances traveled versus number of steps (n) in a random walk, for $n = 1, 2, 3, \dots, 50$, appears to have the shape of the square root of n .

Thus, the concept of random walk with distance traveled was familiar to the COSC/MATH 201 students. A second ABM tutorial led the students through a random-walk simulation, not with one walker, but with up to 1000 walkers, which the user designates with a slider bar (Figure 4). Moreover, the students plotted the average distance traveled by the walkers and the square root of the number of steps (n) versus time, where each walker takes one step per time unit (Figure 5). Multiplying the square root of n by an appropriate constant, about 1.1, results in an empirical model for the data. Reducing the number of walkers to a small number, while observing the walkers, the students can view the more erratic behavior of the graph of the average distance traveled.

```
; Random walk simulation with from 1 to number-of-walkers walkers,
; with plot of the average distance walked from the starting point

turtles-own [x0 y0 turtle-distance]

; setup patches, walkers, and clock
to setup
  clear-all
  ask patches [set pcolor white]
  setup-walkers
  reset-ticks
end

; set up number-of-walkers at random locations near the origin and
; save each walker's starting location in x0, y0
to setup-walkers
  create-turtles number-of-walkers
  [
    set color random 100
    set xcor random 10
    set x0 xcor
    set ycor random 10
    set y0 ycor
    pd
  ]
end

; at each iteration, have each turtle move to a neighbor and
; save its distance from its starting location in turtle-distance
to go
  ask turtles
  [
```

```

    move-to one-of neighbors
    set turtle-distance distancexy x0 y0
  ]
  tick
end

```

Figure 5: NetLogo code for random walk

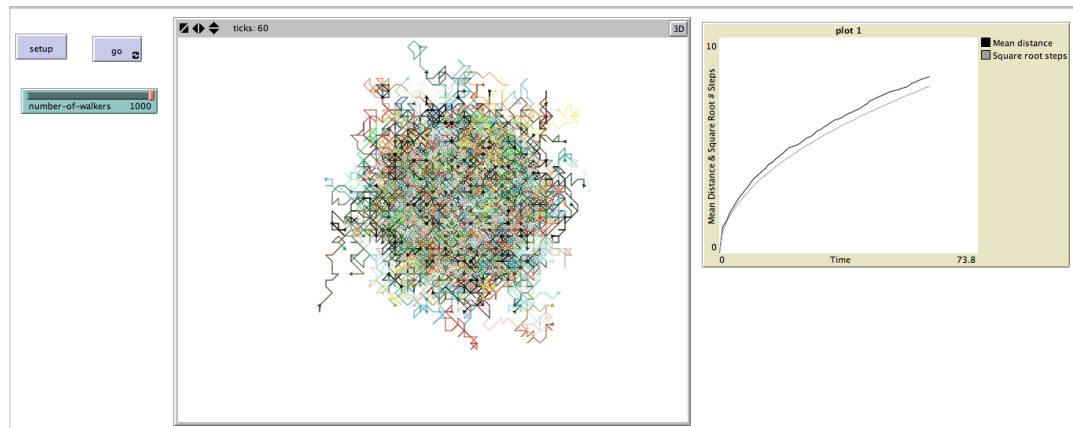


Figure 6: Interface level after one simulation of random-walk code in Figure 4

4 Outcomes

4.1 Pretest

Before covering any material involving ABM and how this technique relates to other methods considered earlier in the semester, the students completed a pretest the last Monday of classes. The pretest involved indicating all the following characteristics that apply for system dynamics models, empirical models, and cellular automaton simulations:

1. affords a global view of major classifications that change with time
2. can be effective in modeling dynamic, spatially complex situations
3. can visualize the progress of events through informative animations
4. computational representation that helps to explain a situation
5. considers the changing sizes of complex interrelated groups of entities
6. consists of a function that fits the data
7. each entity modeled as an autonomous, decision-making agent that has state & behaviors
8. employs matrices to track the simulated behavior of individuals in a community
9. for each time step, proceeds through each entity, revising its state
10. for each time step, sweeps through every cell of the grid, updating its state
11. individual actions and local interactions can impact the whole system
12. models reality with one-, two-, or three-dimensional grids that change with time
13. presents a local view of individuals affecting individuals
14. transition rules that specify the relationship of a cell with its neighbors determine the state of the cell at the next time step
15. updates the states of all the cells in a grid at each time step
16. used to predict behavior where data do not exist

17. world is represented as a rectangular grid of cells, where each cell has a state that can change with time according to set rules

4.2 Posttest

The posttest, administered on the last day of classes after students had completed both tutorials, employed the same pretest questions for modeling techniques, with the addition of a category for ABM.

4.3 Class Performance

Working a self-reported average of about two hours (30 minutes in class, 1.5 between classes), 28 of the 31 students completed Tutorial 1 successfully, while 27 students submitted Tutorial 2. All students completed a significant amount of the tutorial work in class, but three students did not submit their work for either assignment.

Besides gaining an appreciation of agent-based modeling, comparison of the pretest and posttest results showed an increased student understanding of other modeling techniques. Although answers to the pretest questions were not specifically discussed, grades on the system dynamics modeling characteristics rose from an average of 60.7% to 75.3%, and grades for empirical modeling changed from 70.4% to 81.4%. Averages for cellular automaton modeling, which is very similar to agent-based modeling, remained fairly steady, rising from 65.9% to 66.6%; and the agent-based modeling average was similar, 66.1%. These results are based on the submissions of 25 students who completed the two tutorials, the pretest, and the posttest.

4.4 Explorations of Models

Besides working through the tutorials, students submitted paragraphs, one due Wednesday and one Friday, about models, perhaps related to their majors, from the NetLogo Models Library. Models explored included termites, grain growth, team assembly, AIDS, frogger game, rebellion, aggregation of hex cells, Maxwell's demon, crystallization, predation, nuclear fission, illusions, division, movement of moths, tumor, muscle development, 3D surface, binomial rabbits, tetris, rugby, lunar lander, adiabatic piston, fireworks, autumn, wandering letters, followers, ants, membrane formation, herding, and diffusion.

Comments often revealed an exploration of the model, an examination of the code, and a deeper understanding of the results than the instructor had expected. Several critiqued the models and suggested refinements and related models. Besides a number of enthusiastic remarks about the models and results, the following remarks were particularly insightful:

- I was excited when I saw that there was a Firefly synchronization model because we studied the exact same system in Advanced Differential Equations this year. It was very interesting to see a visual representation of what I'd studied mathematically.
- Again, having already taken Thermodynamics in the Chemistry Department, I found this model very interesting. We spent a great deal of time discussing energy transfer processes, and specifically adiabatic ones. For this reason, I found it helpful to be able to visualize an adiabatic process. While we had previously discussed the theory behind it, and the applications of it, I had never been able to see a simulation of the process actually happening.
- Looking at a larger program helped ... give me a better overall understanding of the language.
- ...it was very interesting the factors they considered.
- A model such as this could be used to describe moth behavior around light sources and can be used to predict how they will react to certain light stimuli in nature.

- It's very interesting to see illustrated that the molecules would arrange themselves so with no push but their inherent properties.
- It is an interesting model that shows how a word processor might keep text aligned correctly while the user is modifying other aspects of the document.
- This is an interesting method of showing division in an animated way.
- I really liked this simulation because it shows the math behind something that most of the time is not thought of to be math.
- It is interesting how a rather short code can create a relatively complex image. I am now intrigued and would like to see how other 3D objects are made!
- It was also more interesting to observe the program this time because I know more about NetLogo now so I can better understand what is actually happening in the program and use it.

4.5 Evaluations

On the last day of class, after completion of the posttest, students answered a questionnaire (0 to 5, from strongly disagree to strongly agree) with the following questions and averages:

- The tutorials were clear: 4.0
- I enjoyed learning about agent-based modeling: 4.0
- After this week, I have a better understanding of how the different modeling techniques we have covered compare: 3.3
- I enjoyed exploring some of the NetLogo models on my own: 3.8

Encouraged to make additional comments and suggestions, several students referred to a needed improvement, which we have subsequently made, in one of the quick review questions. One student would have given 5 to better understanding of how the different modeling techniques compare "if we had gone over the answers." A valid suggestion, future coverage of this material will include such a discussion. Other comments included the following:

- It was interesting. Do more with NetLogo.
- Very interesting content. I wish we could have spent more time on it.
- These tutorials were probably my favorite to complete!
- I enjoyed looking at the built-in simulations.
- Really enjoyed working with NetLogo software. Programming one project with it would have been interesting. Possibly cut a week off Stella [system dynamics modeling] in order to add a project with NetLogo.

5 Conclusions

As evidenced by tutorial completion, improved scores on tests of modeling characteristics, reports of existing models, and questionnaire answers, the exploration of agent-based modeling at the end of an undergraduate course in modeling and simulation was very successful. The week accomplished exposing the students to another major simulation technique that is having increasing utility in the natural, physical, and social sciences and served as a conduit for comparison of various modeling and simulation techniques.

References

Buchanan, Mark (2008). "This Economy Does Not Compute," Op-ed, *New York Times*, October 1, 2008.

"Building a Science of Economics for the Real World" (2010). Committee on Science and Technology, Subcommittee on Investigations and Oversight, U.S. House of Representatives (2010). July 20, 2010. Serial No. 111-106.

Chan, Wai Kin Victor, Young-Jun Son, and Charles M. Macal. "Agent-based simulation tutorial-simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation." In *Proceedings of the Winter Simulation Conference*, pp. 135-150. Winter Simulation Conference, 2010.

ECS, Emphasis in Computational Science at Wofford College (2013). <http://www.wofford.edu/computationalscience/> (accessed November 29, 2013)

Hearing of the House Committee on Oversight and Government Reform, "The Financial Crisis and the Role of Federal Regulators," Oct. 23, 2008, preliminary transcript, p. 16, 37 <http://oversight.house.gov/images/stories/documents/20081024163819.pdf>

Hoffer, Lee D. (2013). "Unreal Models of Real Behavior: The Agent-Based Modeling Experience." *Practicing Anthropology* 35, no. 1: 19-23.

Introduction to Computational Science (2014). Website associated with *Introduction to Computational Science: Modeling and Simulation for the Sciences, 2nd Edition*. <http://wofford-ecs.org/IntroComputationalScience/index.htm> (accessed December 15, 2013)

Parker, Jon, and Joshua M. Epstein (2011). "A distributed platform for global-scale agent-based models of disease transmission." *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 22, no. 12. <http://economicsofcontempt.blogspot.com/2009/10/newspaper-front-pages-during-financial.html>

Shiflet, A. and G. Shiflet (2008). *Introduction to Computational Science: Modeling and Simulation for the Sciences*, Princeton University Press

Shiflet, A. and G. Shiflet (2014). *Introduction to Computational Science: Modeling and Simulation for the Sciences, 2nd Edition*, Princeton University Press

Smallwood, Rod, Jenny Southgate, Sheila Mac Neil, Mike Holcombe, Rod Hose, Richard Clayton, Rob Gaizauskas (2013). *Integrative Biology - the Epitheliome Project* http://www.dcs.shef.ac.uk/~rod/Integrative_Biology.html (accessed December 14, 2013)