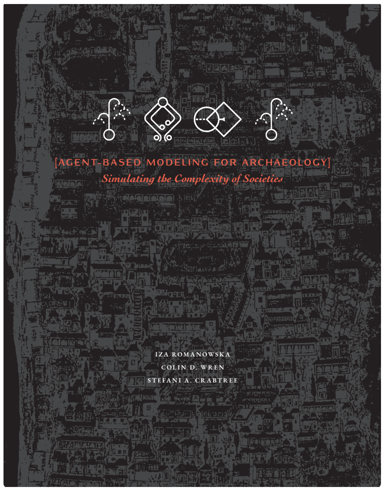
PLEASE NOTE:

The contents of this open-access PDF are excerpted from the following textbook, which is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License:

Romanowska, I., C. D. Wren, and S. A. Crabtree. 2021. *Agent-Based Modeling for Archaeology: Simulating the Complexity of Societies.* Santa Fe, NM: SFI Press.

This and other components, as well as a complete electronic copy of the book, can be freely downloaded at https://santafeinstitute.github.io/ABMA



REGARDING COLOR:

The color figures in this open-access version of *Agent-Based Modeling for Archaeology* have been adapted to improve the accessibility of the book for readers with different types of colorblindness. This often results in more complex color-related aspects of the code than are outlined within the code blocks of the chapters. As such, the colors that appear on your screen will differ from those of our included figures. See the "Making Colorblind-Friendly ABMs" section of the Appendix to learn more about improving model accessibility.

SFI PRESS
THE SANTA FE INSTITUTE PRESS 1399 Hyde Park Road, Santa Fe, New Mexico 87501 | sfipress@santafe.edu

CONCLUSION

Agent-based modeling has become increasingly common in research about human systems. In this book we aimed to bring ABM to all researchers studying the past, and to archaeologists in particular. However, looking from a wider research perspective, learning to use and develop ABMs teaches much more than just a specific simulation technique. We hope that time spent designing, coding, running, and interpreting a diverse array of models has shown you that formal modeling is an incredibly powerful way of gaining insight into the lives of people, in the past but also today. It is also unique because no other method allows us to enter into the complexity of past human intentions, behaviors, and decisions with so much ease while retaining scientific rigor. Many archaeologists aim to understand and explain why, how, and what has happened in the past. But hours spent considering whether to add an extra parameter, which algorithm is the most appropriate, or what value to use illuminates that it is simply not feasible to use pen and paper to unravel the mechanisms driving complex systems. Whether you continue building agent-based models or not, we hope that this volume has armed you with critical tools to evaluate the robustness of our conjectures about the past.

Learning ABM teaches not only a specific simulation technique but model-based thinking.

Throughout this book we have used dozens of models developed both by other archaeologists and by scientists from fields outside of our discipline. We hope that this demonstrates the most important aspect of formal modeling: its cumulative nature. Modeling enables scientists to build *on*, not just *in parallel to*, previous research. This is only possible thanks to the rigor, transparency, and replicability of models described in a formal manner. While in centuries past, data-free narratives that sought to explain causal mechanisms for the trajectory of humanity may have been the norm, this is (thankfully) no longer the case. We must now systematically question these theories and test them against data. Through agent-based models (as well as other formal approaches) we can systematically examine theories, refine our understanding of the past, and advance our scientific practice, for example, by focusing data collection in specific directions.

Decisions involved in developing a model constitute the true scholarship. Coding is just a technical skill.

By now you have probably realized that coding is, in fact, the easiest part of the model-development process. At the beginning it seems like a major hurdle, but with practice you learn how to express your ideas in code with relative ease. What remains challenging are the decisions involved in developing a model: What is important and what can be simplified? Are we representing the key dynamics? Are our parameter values reasonable? How can you compare the results to the archaeological data? Can we justify all of the coding decisions, or would there be unintentional consequences of the way we coded an algorithm? These doubts will likely remain long after the NetLogo debugger stops reporting yet another missing bracket. This is because it is model building and the research process around it that requires real scholarship. For those challenges, unfortunately, there are no easy solutions other than to adhere to the general rules of scientific practice. However, the further we get with our models, the more of the theory space we can cover, ensuring we know more about the systems we study and making it incrementally easier to determine the most plausible description of what the past looked like and what this tells us about people in general.

A Model-Based Learning Trajectory

We covered three aspects of simulation and the process of agent-based modeling for human systems. The first was the epistemological basis of modeling and its place in the canon of scientific practice: What is a model and how does it differ from data? Why do we need to simulate complex systems? What is validation? These are concepts and methods fundamentally important for any scientific pursuit.

The second task was more practical: learning to write a simulation in NetLogo. This included simple aspects like syntax and vocabulary, as well as introducing aspects of computational thinking. Computational thinking refers to how computer scientists approach a problem and the strategies that help them solve it. This includes breaking down a topic into its constituent parts, looking for existing theoretical frameworks and other models to depart from, simplifying and distilling the problem until it is solvable, or even just looking for help on the internet when you get stuck.

Finally, the third goal was to provide a comprehensive overview of existing models in archaeology and related social sciences. The field of complexity science in particular provides us with an extensive range of models spanning multiple disciplines and demonstrating deep regularities among complex systems. Other disciplines that center on humans—economics, demography, sociology, epidemiology—have thought long and hard about how individuals and communities function today. As science is a cumulative process, it is through building on these theories and models and combining them with an understanding of the archaeological data and methods that a modeling approach enables a better understanding of past human societies.

Nevertheless, this book should inspire, not limit, creativity in creating new algorithms and exploring topics in new and unique ways. With the extensive library of algorithms in part II, we hoped to create a steppingstone and a quick reference guide for other modelers to modify, extend, or completely rewrite these models.

How to Build a Good Model

Any formal model is better than an informal model in that it is unambiguous, its assumptions are explicit, and it can be rigorously tested, but some models are better than others. Throughout the book we wove in general advice on creating models and defining modeling research projects that are scientifically robust and informative. We discussed each stage of the modeling process and the key recommendations for best practices. Here we will briefly review the most important points.

Formal models are unambiguous and have explicit assumptions, so they can be rigorously tested.

1. **Research questions**

   Not all research questions are appropriate for a simulation, since many can be addressed simply by analyzing available data. It is worth reminding yourself of the three main functions of simulation: theory building (the "if–then" questions), hypothesis testing ("Was x caused by y or by z?"), and data exploration ("Can x generate data pattern y?"). If your questions consider "why" or "how" or "which hypothesis," then they likely fall into one of these categories. Research questions are critical to science, and equally critical for any modeling study—they guide how you approach building a model, can help you determine what to include and what to exclude, and form the basis for the majority of coding decisions. Research questions often change as we gain new insights from our models. What is important is to adapt them and be clear on what we are trying to achieve.

2. **Ontology building**

   Research questions drive ontology building. They determine what the core model is and which scales are the most appropriate for inquiry. A model is always an abstraction—a simplification of the real system—which means that multiple representations of the same system may be equally valid. Deciding what goes into the model and what does not is challenging, but well-formulated research questions facilitate this task enormously. Perhaps you will base your model on an existing framework, such as building on a published model. Perhaps you will develop your model from scratch, building on your own theories and research. Either way you will need to make *careful* and *intentional* decisions as to how to represent the system you study. Find the most parsimonious way to explore your questions. Don't forget about the trade-offs among realism, generality, precision, and tractability—while you can't have all of them, you can find the right balance for your research questions. In this phase of the modeling process you often realize both how little you actually know about the system you want to study and how multidimensional it may be. Formal modeling is a good strategy for identifying areas of ignorance; while it may not give the answer you want every time, it will at least clarify what you need to learn first.

A model is always an abstraction—a simplification of the real system.

3. **Coding and testing**

   Computer code is a type of language. You can express everything you want in it, as long as you know the vocabulary and grammar rules for how you want to express your ideas. As with any other language, greater practice equates with greater fluency. Nevertheless, even the most experienced computer programmers look for help online simply by searching key terms or reaching out to fellow coders. Don't waste your time figuring out how to solve problems others have already solved; employ Stack Overflow, Google, the NetLogo Users group, and various other boards to find out how others have solved similar problems.

As with languages where you can express similar ideas via different phrases (e.g., I am hungry, I need to eat, I require sustenance), you can code the same process in multiple ways, often producing slightly different results. Try different methods to ensure that the results are not a direct consequence of the particular algorithm you used.

Don't waste your time figuring out how to solve problems others have already solved.

Testing your code is a must. It is your responsibility to ensure that the results you report can be trusted. Do not depend on your perception or intuition alone—write tests in the code that will catch problems where you may not have suspected them. Optimizing the code is useful, but remember the computer-science motto: "premature optimization is the root of all evil." The correct sequence of actions is to write code, make it work, make sure it works correctly (test it), optimize it, and then test it again. Take seriously your responsibility for having code that does what you think it does.

4. **Parameterization and experiment design**

   It's critical to ensure that your model doesn't fall into the category of "garbage in, garbage out." Your model *will* produce results, but you must be careful with algorithms and parameter values and ensure they are representative of the phenomenon you are modeling. A common misconception of an agent-based model is that you can make it produce any result you want. To an extent this is true, but only for very poorly constructed models. Whether it is regression models or spatial statistics or any other type of modeling, if you manipulate data and parameters enough you can create misleading results. Following general scientific practice shields you from these kinds of mistakes. To avoid these problems, begin with strong research questions, adhere to principles of parsimony, and then run a sensitivity analysis to refine your understanding of which parameters you need to be particularly careful with. Give yourself enough time to fully explore the dynamics of your model. This can easily take 40–50% of the project's time.

5. **Validation, analysis, interpretation, and publication**

   It is *critical* not to use the same data for input as for validation. This is not the only challenge in validation, however. Be mindful of biases in archaeological (or other) data and their potential impact on your interpretation. Attempt to fully understand your data before comparing it to your modeled results.

   If we have been successful in convincing you to switch to a scripting language (R, Python) for your data analyses, many of the axioms discussed here hold true there as well. Best practices for writing code in an agent-based model are general best practices for writing all code—make sure you test your data analysis scripts.

   Scientific reproducibility means that everything researchers produce should be transparent and accessible (Marwick et al. 2017). You may be concerned that your code is poorly written or contains bugs, but the ethos of programming is always to share your code because in the end it makes for better science. Likewise, be kind when you review your colleagues' code. If they made a mistake, they probably did not do it on purpose—it should not negate the months of work they put into their simulation—and you may be on the receiving end next time.

   Finally, keep in mind the audience: archaeologists, historians, social scientists, and other members of the public with a fascination for human societies. The goal is not to write a simulation but to refine our understanding of human societies. When you present your results, be mindful of your audience, and discuss the model and results in a way that enables nonmodelers to fully appreciate the significance rather than hide behind technical jargon.

The goal is not to write a simulation but to refine our understanding of human societies.

Learning to Be a Good Modeler

Becoming a good modeler goes beyond learning how to code. Rather, learning to be a good modeler requires constantly learning, adapting to new techniques, and maintaining a healthy dose of critical thinking. While we have touched on all of these points throughout the text, here we review the main elements in striving to be a good modeler.

1. **Learning continuously**

   Code development is not static. When you learn a language, you are never done learning it; there will be new developments, new primitives, new packages, and new ways of building models and exploring your results. You must stay up-to-date on platform development, updating your versions of NetLogo, R, or Python. If a dramatically new version of the platform is released, test to ensure that your older models still work. Often, you'll need to update a few primitives to make sure the model still runs. This can be a good way to learn the improved functionality in the new version, and it also helps others who may want to use your models. Equally, the more you code, the more you realize that there are multiple ways to achieve the same end goal. However, some of them are better or more appropriate in certain situations. This is what computer scientists mean when they talk about "elegant" code—it may not do anything beyond a different, more messy implementation, but it can be more computationally efficient or more readable.

Remember, code development is not static. When you start learning a language, you are never done learning it.

2. **Combining methods in new ways**

   One of the most difficult tasks of becoming a proficient agent-based modeler is that you need to embrace the idea of becoming a polymath. While academia may desire depth in a specific field, modeling requires breadth to be able to include in your model theories, frameworks, and algorithms from other disciplines. Modeling a past society may require models of trade, social cohesion, foraging theory, and network theory, requiring you to read up on economics, sociology, behavioral ecology, and physics—all on top of developing strong computer-science skills. Talking to experts in other departments, reading new books and articles when they come out, and constantly trying to acquire knowledge outside of your home discipline will enable you to ask and answer more interesting questions. Becoming proficient in finding theories, data, and models from other fields, adapting them into code, and then examining the results will go a long way toward advancing a more transdisciplinary and synthetic science.

3. **Making mistakes and learning from them**

Frequently, we learn as we go, and we need to become adept at "failing gloriously."

The challenge of the juxtaposition of depth as a traditional scholar with the breadth that is required to be a successful agent-based modeler is that often you will not get things right on the first, second, or even third try. After you add new code and click NetLogo's CHECK button, the debugger will flag errors more often than not. After fixing those errors, running the model tests will catch other errors by failing some more. Later, the peer reviewers will ask uncomfortable questions, which may expose hidden errors. Finally, a student may drop you an email years later saying they failed to replicate your model's results. Frequently, we learn as we go, and we need to become adept at "failing gloriously" as described by Graham (2019). Graham recommends not just tinkering and breaking things as a way to discover, but intentionally being open about our struggles and successes with coding. Computational social science and digital humanities are relatively new fields, which can lead to opportunities as well as setbacks. Learning to embrace the challenge of having things not work out the way you plan for your models—whether they don't have the results you expect, you can't find the data set you need, or you realize you can't figure out how to code the problem you want—is key for pursuing this type of work. Failing gloriously means normalizing "the messiness of doing digital work" so that others will try as well (Graham 2019, pp. 77–78). Relying on the community of other modelers and learning from your mistakes and the errors of others advances science for all of us.

4. **Employing model-based thinking**

Once you begin writing agent-based models to formally assess theories about the past, you may find that *model-based thinking* infiltrates the way you see the world. Model-based thinking can help you think more critically and creatively; not only can models help you explore the past in more depth, but understanding the ways that different processes can interact is also useful for understanding the world around us. Several of the authors of the original *Artificial Anasazi* model brought their skills in agent-based modeling, honed on an archaeological case study, into diverse fields like economics and public health. This ability to employ model-based thinking may lead you to question reports of breaking studies, or even the published study itself—for example, when the results of the study are said to "prove" something definitively or they "infer" causation based only on the correlation of two factors in a complex system. Because you have spent time learning to build models, to test models, and understand principles of equifinality, you now know a model can point to a likely result, but will not definitively solve an issue. New data and new models can build on past data and models, refining our understanding. Being a modeler means you can always employ your toolkit to build simple models to seek a better understanding of current and past processes.

Some senior scientists have suggested that learning to code is easy, but learning the intuition of what makes a good model is hard. This is true. But applying the lessons learned throughout this book and adapting these techniques into your own scientific practice means you can build up that intuition. If more people knew how to apply these approaches to day-to-day life, it would help our society learn how to approach questions on complex systems.

Learning to code is easy, but developing an intuition for what makes a good model is hard.

5. **Being prepared for the challenges of interdisciplinarity**

If you enter the modeling pathway, you'll quickly find that it opens a whole new realm of career options while also making some options more challenging. Without any doubt your skills will be in high demand, and your interdisciplinary profile can open academic worlds parallel to archaeology. At the same time, you may find yourself neither here, nor there, balancing the requirements and expectations of very different disciplinary audiences. There are significant systemic barriers to researchers with interdisciplinary credentials (e.g., grants, positions, or publications are judged by disciplinary panels), but remember that simulation is a popular tool not only across academia but also outside of it.

Final Thoughts

Agent-based modeling of archaeological systems can help us quantify and understand the trajectories of past societies, but it can also do much more than that. The saying "the past is a foreign country; they do things differently there" (Hartley 1953) could equally be rephrased as "the future is a foreign country." By using the past as a calibration dataset, we can better understand where we are today, and where we are going tomorrow. The past is a powerful tool for examining how individuals and groups react in a plethora of different situations. In this way, it can be seen as a set of already conducted "experiments," showing us possible solutions to the challenges societies face. These experiments are all we have to go on as we attempt to predict the trajectory of our future.

By using the past as a calibration dataset, we can better understand where we are today, and where we are going tomorrow.

Ultimately, what connects the past, present, and future are people: the shared set of abilities and propensities that make us all human. By modeling people of the past, we learn more about humans and the different ways in which we negotiate our lives. Thus, the tools in this book that focus on gaining understanding of how ancient societies conducted their business are useful for archaeology, but also make archaeology useful beyond our disciplinary boundaries. While many will argue archaeology does not need to be useful, as it simply pursues knowledge about the past, with a systematic method of testing and developing our theories, we could bring archaeology closer to the forefront for policymakers and have an impact on modern society.

In the end, it is our goal that the skills you have developed—of model-based thinking, building on theories and models of other disciplines, developing clear representations of your hypotheses and testing them in rigorous ways, as well as following open-science principles—will lead to a growth in quantitative archaeology and computational social science. We hope that critical evaluation of even complex theories and hypotheses using formal tools will become a norm in our discipline. One turtle at a time.

Acknowledgments

We would like to thank many of our friends and colleagues who have contributed code, articles, suggestions, and reviews for their tireless work. We are particularly indebted to those who read drafts of these chapters over the past few years: Claudine Gravel-Miguel, Arthur Hjorth, Ben Davies, Shawn Graham, Stan Rhodes, Stef Boogers, Ross Hammond, Kyle Bocinsky, Phil Riris, James Allison and his Archaeological Science students, Fulco Scherjon, Tom Brughmans, Enrico Crema, Isaac Ullah, peer reviewers of this volume, and the hundreds of students, early and not-that-early career researchers, who participated in our classes, workshops, and summer schools, providing feedback in real time. We would also like to acknowledge those who taught us to model: Seth Bullock, Jason Noble, Hans Fangohr, Luke Premo, Mark Lake, Andre Costopoulos, Anne Kandler, Julian Z. Xue, Tim Kohler, Ben Ford, Denton Cockburn, and all the classmates who learned alongside us and whose input dramatically improved our own abilities. We are also indebted to David Krakauer and the SFI Press team—Sienna Latham, Laura Egley Taylor, Katie Mast, and Bronwynn Woodsworth—who worked tirelessly to make this book vastly better than when they first agreed to publish it.