

Some Thoughts on Agent-based Modeling and the Role of Computation in Economics

By W. Brian Arthur,*

7 June, 2021

Science changes as its methods change: each new method (think of the telescope, the microscope, the mathematics of crystallography) brings a new mode of understanding and changes what we see. In economics methods have changed many times, as geometry, algebra, calculus, and statistics have been introduced. Each of these has provided a new mode of understanding and each has changed economics in its turn. Now computation is entering economics and is changing our understanding, largely in the form of agent-based modeling. In this paper¹ I want to discuss how this is happening and ask three questions about it. What is the relation between complexity economics and computation, in particular agent-based computational modeling? What advantages does computation bring to economics? And does computation have a wider role in economics than agent-based modeling?

Complexity economics and agent-based modeling

Standard, neoclassical economic theory views agents in the economy (producers, investors, banks, government agencies) as rational decision makers facing well-defined problems and reacting with optimal behavior. To make this amenable to mathematics, it assumes the agents are representative (fall into one or a small number of types) and have knowledge of these types, and that their behavior is in equilibrium with—consistent with—the outcomes this behavior creates.² This approach has worked well, but over the years economists have felt its assumptions were not always realistic, and have sought a different approach (see Simpson, 2013).

Complexity economics, which started largely at the Santa Fe Institute in the late 1980s, relaxes these assumptions and seeks a more realistic way to look at the economy (Arthur 1999, 2015a, 2021a). It assumes more generally that agents in the economy differ and that their experience and resources and circumstances differ. This immediately brings up a problem. If we admit the reality of heterogenous agents or firms, we open to a situation of fundamental uncertainty: agents in general do not know what other agents will do and cannot assign probabilities to this. In this circumstance, the problem agents face becomes ill-defined, which means that a “logical response” becomes ill-defined, which means that “rationality” itself becomes ill-defined. It ceases to exist. The standard theory comes to a halt.

People do however act within situations that are ill-defined. Typically they generate hypotheses or ideas about the situation they are facing, try to make sense, explore, try out ideas and discard responses that don’t work, and occasionally they come up with novel responses. These reactions in turn change the situation or outcome, so that agents may have to react or adjust again, and no equilibrium need necessarily emerge. In certain cases it may be possible to

* Santa Fe Institute, PARC, and Stellenbosch Institute for Advances Studies (SIAS).

model this using algebraic equations, but more generally these complications force models to be computational, and this is where agent-based modeling comes in. Models rely heavily on agents' behavioral responses and so we think of them naturally as agent-based.

The thinking behind agent-based modeling needn't necessarily derive from complexity economics, though often it does. Indeed some of the very early agent-based models, the El Farol Bar problem (Arthur 1994), the Santa Fe Artificial Stock Market (Palmer *et al*, 1994; Arthur *et al*, 1997), came out of complexity economics. Whatever their basis, agent-based models have become widely popular in economics (for overviews see Tesfatsion & Judd, 2006; Axtell and Farmer, 2022; Hommes and LeBaron, 2018). Indeed, an approach related to complexity economics is called agent-based computational economics [ACE] (Tefatsion 2006, 2021a, 2021b). Complexity economics and ACE overlap considerably.³ Where complexity economics starts from a theoretical viewpoint and arrives at agent-based computational modeling, agent-based computational economics starts with a method, agent-based modeling, and uses complexity thinking as a backdrop to this. Whichever label we use, if we venture beyond neoclassical assumptions and admit significant heterogeneity or fundamental uncertainty or nonequilibrium, we require computation to arrive at outcomes.

What can agent-based computational models provide?

Computational modeling (I'll speak in this section mainly of agent-based modeling) is still relatively new in economics. What does it provide that is worthwhile? And what can it show us that is different? If we compare it to the standard equation-based approach, the two traditions have much in common.⁴ Both approaches trace a pathway from agent behavior to its implied outcome. But each has different advantages. Equation-based modeling allows us to reach conclusions exactly and reliably based on clearly stated assumptions, following the logic step by step and thus understanding what it reveals. Computational modeling lacks this advantage, but it compensates by allowing more precise modeling.

This may seem counter-intuitive. Usually we think of equation systems as exact and computation as *ad-hoc* and inexact. But the exactness of equations is an illusion. "As soon as you write an equation it is wrong," said physicist Albert Tarantola, "because reducing a complex reality to an equation is just too simplistic a view of things" (quoted in Bailey 1996, p.29). In fact, much of mathematics-based economic theory consists of pointing out how averaged things (unemployment, say) affect averaged things (inflation, say), which yields at best coarse-grained theory. This hides a lot of what is really going on. A production function links average inputs to average outputs, but it tells us nothing about the technology or detailed organization or means by which this all-important transformation happens. Standard economics adopts such idealized assumptions because equation systems need to stay simple to be mentally manageable. Algorithms by contrast can be expanded to encompass an arbitrary amount of realistic detail. They can thus tell a more nuanced story and avoid mean-field approximations. Where the inclusion of realistic details matters algorithms are more accurate.

In theory then, a computational approach can take us beyond an equation-based one with its forced simplicities. It is useful to regard computational models as laboratory experiments, running these to see what they reveal, exploring what phenomena appear and the conditions under which they appear. Models can be set up as carefully-designed computer experiments, and we can use these to isolate phenomena, explore phase changes and the mechanisms that cause them. Sometimes we can construct simpler toy models of a phenomenon that capture its essential features and allow us to use mathematics or stochastic theory to study it. The objective here, whatever method we use, is to obtain general insights. Rigor in a computational setting

needs to widen from correctness of the logic that connects assumptions to their implications, to insistence on honest, realistic modeling with reproducible, verifiable results.

§

One test of computation in economics is whether it can demonstrate or explain phenomena that the standard equation-based framework cannot. Are there findings that come from computational modeling the standard approach can't easily see?

I believe there are. Let me give one example, the theory of asset pricing (or stock market pricing). The classic equation-based approach (Lucas 1978) assumes identical investors using identical forecasting models that are on average statistically validated by the prices they forecast. This rational-expectations approach works well to explain how market prices form how they reflect the sequence of random earnings, and it has been much lauded. But it has some key failings. In this imagined market no trade takes place at all. This is because the investors are identical. If one of them wants to buy, all want to buy and there are no sellers to make the trade; if one wants to sell, all want to sell and no buyers exist. The stock price simply alters to reflect changes in demand. Also, the theory cannot account for real market phenomena such as the emergence of a market psychology, or price bubbles and crashes, or periods of random high and low price variations (volatility), or the emergence of technical trading (trades based on the recent history of price patterns).

The Santa Fe artificial stock market model, an early agent-based computational model set up in 1988, works differently (Palmer *et al*, 1992; Arthur *et al*, 1997). In this model our 'investors' or agents are small but intelligent computer programs that can differ from one another. These share no self-fulfilling forecasting method, but are required somehow to learn or discover forecasting methods that work. Our investors randomly generate their own individual forecasting methods, try out promising ones, discard methods that don't work and occasionally generate new methods to replace them. They get smarter as they do this. They make bids or offers for a stock based on their currently most accurate methods and the stock price forms from these—from our investors' collective forecasts.

This computer experiment, when we ran it, showed two regimes or phases. At low rates of investors' exploring, the market behavior converged into Lucas's rational-expectations equilibrium. Investors' forecasts are "attracted" into becoming alike and so trading fades away. Here, the neoclassical outcome holds. But if our investors try out new forecasting methods at a faster, more realistic rate, the system undergoes a phase transition and a complex regime emerges. The market develops a rich psychology of different beliefs; a robust volume of trade emerges; price bubbles and small crashes appear; technical trading emerges; and random periods of volatile trading and calm trading emerge.

In other words, phenomena we see in real markets emerge⁵.

We see this last phenomenon of random periods of high and low volatility because if some of our investors occasionally discover new profitable forecasting methods, they will invest more. This alters the market slightly, and so other investors also change their forecasting methods and their bids and offers. Changes in forecasting beliefs in this way propagate through the market in avalanches of all sizes, and this causes periods of high and low volatility.

I should point out here that phenomena such as technical trading, or bubbles and crashes random volatility, are not "departures from rationality." Outside of equilibrium, "rational" behavior is not well-defined. These features arise from agents discovering behavior that works temporarily in situations caused by other agents discovering behavior that works temporarily. This is neither rational nor irrational, it merely emerges.

A deeper role for computation in economics

So far I have been discussing computation in economics mainly as agent-based modeling. Now I want to talk about a wider role for computation in economics.⁶ In what follows I will talk about standard mathematics or “algebraic mathematics,” where I mean standard algebra, linear algebra, and calculus as used in economic modeling. And I will switch from talking about “computation” to talking about algorithms. I will borrow heavily from Algorithmic Information Theory, as developed by mathematician Gregory Chaitin (1990, 2006) and others.

Algorithms are methods, they are specified processes. Computations are narrower; they are operations that happen when we “compute out” the numerical consequences of some set of instructions. In this sense Gale and Shapley’s (1962) famous solution to the college admission problem is an algorithm. It is a general process, instructions if you like, for matching students with colleges; using it in a particular setting is a computation. Similarly Roth et al.’s (2004) organ allocation process for kidney transplants is an algorithm, or method; using it to assign organs is a computation. Both studies offer algorithms as solutions to particular economic problems, but the solutions are abstract processes, not particular computations themselves. Notice that algorithms here are objects to be sought and they stand in their own right. This way of thinking is not new in computer science. Turing (1936) viewed algorithms—he called them “methods”—as mathematical objects in themselves, and used them abstractly in his proofs. Algorithms then become objects in a different form of mathematics—algorithmic mathematics⁷—which we can think of as a language for creating algorithms and “the mind-set of thinking in terms of algorithms for solving problems and developing theory” (Maurer, 1984). This mathematics, as Turing (1936), Chaitin (2006) and Wolfram (2002) have shown us, gives us a different world, one that’s just as logical and “mathematical” as the algebraic one.

We can then say this. Standard modeling uses sets of equations combined logically to model systems; in doing this it draws from algebraic mathematics as a language of expression. Computational modeling uses sets of statements combined logically to express the workings of some system; in doing this it draws from algorithmic mathematics as a language of expression. We have two languages, algebraic mathematics and algorithmic mathematics, as means by which to model systems. These have much in common, but differ in what they can easily express. Algebraic mathematics models quantities balancing quantities in equation systems; and new quantities issuing from previous quantities in differential systems. It excels at modeling relations between quantities.⁸

Algorithmic mathematics excels at modeling processes, or actions if you like.

It does this easily because algorithms themselves are sets of actions: do this, sort that, evaluate this, and so actions enter directly. Actions also enter indirectly. Algorithms allow conditional branching—they contain *if-then* clauses—so we can have descriptions saying: If process *G* and *H* have been completed, and if process *M* has begun, then execute sub-process *K*. Algorithmic modeling allows processes, actions and events, so we can have actions inhibiting other actions, actions triggering other actions, actions calling other actions. This is powerful: it opens a wide range of possibilities in understanding the economy not as a set of quantities—prices, indices, amounts—that move up and down, but as a set of entities—institutions, arrangements, structures, solutions, organizations—being sequentially constructed.

Algorithmic mathematics, then, allows us to tell process-based “stories” that can be far richer than the standard noun-restricted ones: stories about formation and structural change and creation and development. This point seems somewhat abstract, but it is fundamental. The mathematical language we use in a science shapes our understanding in that science (Gisin, 2021); in this case it widens our understanding of how the economy operates.

How do these ideas apply to economics in general? In an earlier paper (Arthur, 2015b) I pointed out that there are two great problems in economics. One is *allocation* within the economy: how quantities of goods and services and their prices are determined within and across markets. This is represented by the great theories of general equilibrium, international trade, and game-theoretic analysis. The other is *formation* within the economy: how an economy emerges in the first place, and grows and changes structurally over time. This is represented by ideas about innovation, economic development, structural change, the role of history, institutions, and governance in the economy. The allocation problem is well understood and highly mathematized; the formation one less well understood and barely mathematized.

An algorithmic approach above anything else allows problems of formation in economics to be studied. Because it allows processes to be included, it stands to show us economic development not as a series of quantity adjustments to GDP or industry, but rather as a set of actions—millions of actions, perhaps—setting up trades and small entrepreneurships, getting education moving, forming new habits of business, deepening banking systems, creating new products and services. We can see development *directly* as a process. More generally, an algorithmic approach can show us how institutions form, and how technological innovation works to transform and re-create the economy. We can then bring these questions into formal economic theory. A process view in economics shows us a world where large and small structures continually form, where agents and organizations continually respond to their internal and external environment, where agents can change from within, where fresh undertakings continually create novelty. In all these senses the economy comes alive.

We still have one point to settle. We accept conventionally that algebraic models constitute theory in economics. Can algorithms also constitute theory? My answer is yes. I will simply summarize the argument (see Arthur 2021b) here: Chaitin (2006), points out that theories are condensed descriptions of how some system works. Newton's equations are thus condensed descriptions of the motions of planets: we can “decompress” or unfold the information contained in them to arrive at elliptical planetary orbits. As such Newton's equation-based system constitutes a “theory” of planetary orbital information. The same can be said for algorithms. If we have an algorithm that says Write two 0s, then two 1s, then repeat, it will produce that sequence 0011 0011 0011 0011 indefinitely. The algorithm is a compression of the information in the computed system, and just as with Newton we can say it constitutes a “theory” of the system. Algorithms used to describe systems—to condense the information in a complicated situation—are theories of such systems. It follows that if economic models based on algorithmic specifications are condensed descriptions of how systems work, we can see them as theory proper, just as we see equation-based descriptions as theory proper.

Conclusion

What I take away from these ideas is that computation in economics means more than simulation or agent-based modeling. It offers economics an alternative mathematics, algorithmic mathematics, within which new solutions can be constructed. Algorithmic mathematics thus stands alongside algebraic mathematics as a new language for theory in economics.

Of course the world that computation reveals is not one of rational perfection, and it is not mechanistic. It is more biological than anything else. Its agents are constantly acting and reacting within an “ecology” of behaviors brought about by other agents acting and reacting. Algorithmic expression allows novel, unthought of behaviors, novel formations, structural change from within—it allows creation. It gives us a world that is constantly creating and re-creating itself. The proper use of computation, I believe, is to explore this realm that lies beyond our current limitations.

There is a danger of course that economics will not open to this world but will stick to its realm of stasis, order, and perfection. But I don't think that will happen. Economics will change if for no other reason than that all the other sciences are changing. They are wondering how structures are generated. They are getting away from static objects to ideas of process. They are embracing uncertainty, ill-definedness, and unpredictability, and dropping their Newtonian insistence on a predictable, ordered world. Economics is opening slowly to this change in *Zeitgeist* and beginning to accept nonequilibrium, ill-defined problems, and processes. More than anything else, computation is catalyzing this change and making it possible. So too is biology, which is a field of processes (speciation, embryonic development, protein expression) responding to and inhibiting other processes. Inevitably economics will open to these changes in the thinking of our times.

References

- Arthur, W. Brian, "Complexity and the Economy," *Science*, **284**, 107-109, 1999.
- Arthur, W. Brian, "Inductive Reasoning and Bounded Rationality: the El Farol Problem," *Amer Econ Rev Papers & Proc.*, **84**, 406-411, 1994.
- Arthur, W.B., Holland, J.H., LeBaron, B., Palmer, R. & Tayler, P. Asset Pricing under Endogenous Expectations in an Artificial Stock Market. In *The Economy as an Evolving Complex System II*, Arthur, W.B., D. Lane, S. Durlauf (eds.) Addison-Wesley, 1997.
- Arthur, W. Brian, *Complexity and the Economy*, Oxford Univ. Press, 2015a.
- Arthur, W. Brian, "Complexity Economics: A different Framework for Economic Thought," in *Complexity and the Economy*, 2015b.
- Arthur W. Brian. "Algorithms and the Shift in Modern Science," *Beijer Institute Discussion Paper 269*, Swedish Academy of Sciences, 2020.
- Arthur, W. Brian. "Foundations of Complexity Economics." *Nature Reviews Physics*, **3**, 136-45, 2021a.
- Arthur, W. Brian, "Economics in Nouns and Verbs." Preprint on ArXiv, <https://arxiv.org/abs/2104.01868>, 2021b.
- Axtell, R. & Farmer, D. Agent-Based Modeling in Economics and Finance: Past, Present, and Future. This volume, 2022.
- Bailey, James, *After Thought: The Computer Challenge to Human Intelligence*, Basic Books, 1996.
- Brock, W.A., Lakonishok, J. & LeBaron, B. "Simple Technical Trading Rules and the Stochastic Properties of Stock Returns." *J. Finance* **47**, 1731-64, 1992.
- Chaitin, Gregory, *Information, Randomness, and Incompleteness: Papers on Algorithmic Information Theory*, World Scientific, Singapore, 1990.
- Chaitin, Gregory, *Meta Math!* Vintage Books, New York, 2006.
- Colander, D. (ed.), *The Complexity Vision and the Teaching of Economics*, Edward Elgar, 2000.
- Gale, David, & Lloyd Shapley, "College Admissions and the Stability of Marriage," *American Mathematical Monthly*, **69**, 1, 1962.
- Galla, T. & Farmer, J.D. "Complex Dynamics in Learning Complicated Games." *PNAS* **10**, 1232-36, 2013.
- Gisin, N. "Mathematical Languages Shape our Understanding of Time in Physics." *Nature Physics*, **16**, 114-119, 2020.
- Hommes, C. & LeBaron, B., eds. *Handbook of Computational Economics, Vol IV: Heterogeneous Agent Modeling*, North-Holland, 2018.

- Hommes, C.H. “Bounded Rationality and Learning in Complex Markets.” In *Handbook of Research on Complexity*, E. Elgar, 2009.
- Kopel, M. “Oligopoly Dynamics.” In *Handbook of Research on Complexity*, E. Elgar, 2009.
- LeBaron, B., Arthur, W.B., & Palmer, R. “Time Series Properties of an Artificial Stock Market.” *J. Econ. Dyn. Control*, **23**, 1487-1516, 1999.
- Lucas, R. “Asset Prices in an Exchange Economy.” *Econometrica* **46**, 1429-45, 1978.
- Maurer, S. “Two Meanings of Algorithmic Mathematics.” *The Mathematics Teacher*, **77**, 430-35, 1984.
- Miller, J. & Page, S. *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*, Princeton Univ. Press, 2007.
- Mirowski, Philip. *Machine Dreams: Economics becomes a Cyborg Science*. Cambridge Univ. Press, 2002.
- Palmer, R.G., Arthur, W.B., Holland, J., LeBaron, B. & Tayler, P. “Artificial Economic Life: A Simple Model of a Stock Market,” *Physica D* **75**, 264-74, 1994.
- Roth, Alvin, “The Economist as Engineer: Game Theory, Experimentation, and Computation as Tools for Design Economics,” *Econometrica*, **70**, 341-78, 2002.
- Roth, Alvin, Tayfun S. & M. Unver, "Kidney Exchange," *Quarterly Jnl. Economics* **119**, 457-88, 2004.
- Simpson, D. *The Rediscovery of Classical Economics: Adaptation, Complexity, and Growth*, E. Elgar, 2013.
- Tesfatsion, L. & Judd, K.L. eds. *Handbook of Computational Economics 2. Agent-Based Computational Economics*. (Elsevier North-Holland, NY, 2006a).
- Tesfatsion, L. “Agent-Based Computational Economics: A Constructive Approach to Economic Theory.” In Tesfatsion & Judd, *op. cit.* 2006b.
- Tesfatsion, L. “Agent-Based Computational Economics: Overview and Brief History,” paper prepared for the *Shu-Heng Festschrift*, forthcoming 2021a.
- Tesfatsion, L. Agent-Based Computational Economics (ACE): Homepage
<http://www2.econ.iastate.edu/tesfatsi/ace.htm> , 2021b.
- Turing, Alan, “On computable numbers with an application to the Entscheidungsproblem,” *Proc. of the London Mathematical Society*, Vol. 43, 1936.
- Wolfram, Steven, *A New Kind of Science*, Wolfram Media, 2002.

Endnotes

- ¹ In this paper I heavily use ideas from previous papers of mine (Arthur 2015b, 2020, 2021a, 2021b).
- ² Other, similar approaches to economics also use computation (e.g. Epstein, 2006).
- ³ ACE was named by Tesfatsion in 2006; complexity economics was named by Arthur in 1999.
- ⁴ Indeed, computational models often contain equations.
- ⁵ Studies by LeBaron et al. (1999), Kopel (2009), Hommes (2009), Galla and Farmer (2013), similarly find regime transitions from equilibrium to complex behavior in nonequilibrium models.
- ⁶ For a wider view see Mirowski (2002) and Roth (2002).
- ⁷ Like standard mathematics, algorithmic mathematics has its own grammar, vocabulary, rules, and habitual structures, and often these are expressed in the form of a programming language, such as Python, or C++. Algorithmic mathematics can include equation systems from standard mathematics, and it includes a large component of logic, in this case Boolean logic.
- ⁸ For a fuller discussion of how standard mathematics is noun/quantity based, how algorithmic mathematics is verb/process based, and the implications of this, see Arthur (2021b).