

WHY AGENTS? ON THE VARIED MOTIVATIONS FOR AGENT COMPUTING IN THE SOCIAL SCIENCES

Robert Axtell*

Center on Social and Economic Dynamics
Working Paper No. 17
November 2000

Abstract

The many motivations for employing agent-based computation in the social sciences are reviewed. It is argued that there exist three distinct uses of agent modeling techniques. One such use — the simplest — is conceptually quite close to traditional simulation in operations research. This use arises when equations can be formulated that completely describe a social process, and these equations are explicitly soluble, either analytically or numerically. In the former case, the agent model is merely a tool for presenting results, while in the latter it is a novel kind of Monte Carlo analysis. A second, more commonplace usage of computational agent models arises when mathematical models can be written down but not completely solved. In this case the agent-based model can shed significant light on the solution structure, illustrate dynamical properties of the model, serve to test the dependence of results on parameters and assumptions, and be a source of counter-examples. Finally, there are important classes of problems for which writing down equations is not a useful activity. In such circumstances, resort to agent-based computational models may be the only way available to explore such processes systematically, and constitute a third distinct usage of such models.

* *Corresponding author address:* Robert Axtell, Center on Social and Economic Dynamics, The Brookings Institution, 1775 Massachusetts Ave. NW, Washington, DC 20036; e-mail: raxtell@brook.edu; web: <http://www.brook.edu/es/dynamics>.

WHY AGENTS? ON THE VARIED MOTIVATIONS FOR AGENT COMPUTING IN THE SOCIAL SCIENCES

Robert L. Axtell
The Brookings Institution*

Abstract The many motivations for employing agent-based computation in the social sciences are reviewed. It is argued that there exist three distinct uses of agent modeling techniques. One such use — the simplest — is conceptually quite close to traditional simulation in operations research. This use arises when equations can be formulated that completely describe a social process, and these equations are explicitly soluble, either analytically or numerically. In the former case, the agent model is merely a tool for presenting results, while in the latter it is a novel kind of Monte Carlo analysis. A second, more commonplace usage of computational agent models arises when mathematical models can be written down but not completely solved. In this case the agent-based model can shed significant light on the solution structure, illustrate dynamical properties of the model, serve to test the dependence of results on parameters and assumptions, and be a source of counter-examples. Finally, there are important classes of problems for which writing down equations is not a useful activity. In such circumstances, resort to agent-based computational models may be the only way available to explore such processes systematically, and constitute a third distinct usage of such models.

THE NEED FOR COMPUTATIONAL MODELS IN GENERAL AND AGENTS IN PARTICULAR

The ideal gas, the perfect fluid, the Eulerian beam — such ideal types are commonplace in science. Each ideal type plays an important role in its field, both pedagogically and practically. Students are taught about ideal types in order to build their intuition about fundamental relationships between key variables. Knowledge of such ideal types is also helpful in making approximate, order-of-magnitude calculations. But reality is not ideal: real gases depart systematically from the ideal gas law, real fluids have viscosity, and real beams buckle in all kinds of ways that Euler never imagined.

Rational agents are an ideal type. They are introduced to undergraduates in order to teach first principles, e.g., strategic behavior, incentives, expectations, substitution effects, moral hazard, adverse selection. To their credit, the undergraduate textbooks are quick to circumscribe the domain of the rational agent. It is something of a curiosity that only in graduate school is this ideal type reified into the main object of study.

Or so it has been up until recently. There are now a variety of approaches to political economy that attempt, each in its own way, to move beyond the rational agent. These include experimental economics, evolutionary economics, and certain computational approaches. This paper is concerned primarily with the latter approach, although it is also relevant to the others

* *Corresponding author address:* Robert Axtell, Center on Social and Economic Dynamics, The Brookings Institution, 1775 Massachusetts Ave. NW, Washington, DC 20036; e-mail: raxtell@brook.edu; web: <http://www.brook.edu/es/dynamics>.

insofar as computational economics makes use of evolutionary ideas and results from experiments.

There is a branch of computational economics in which relaxing the dependence on rational agents plays no important role. This stream of thought is well represented by Amman *et al.* [1996] and Gilli [1996], and concerns itself primarily with efficient numerical solution of equation-based models involving rational agents, or, with bringing new optimization techniques to bear on classical economic problems.

The other main branch of computational economics involves agent-based models. In such models, individual agents are explicitly represented. These agents interact directly with one another, and social macrostructure emerges from these interactions.¹ A very common motivation for such models is, broadly speaking, a basic dissatisfaction with rational agents. Thus, essentially all agent-based models that have appeared to date involve some form of boundedly rational agent.² This paper is concerned with various uses of agent-based models in the social sciences generally and in political economy in particular.

Agent-Based Computation: Strengths and Weaknesses

An agent-based model consists of individual agents, commonly implemented in software as objects. Agent objects have states and rules of behavior. Running such a model simply amounts to instantiating an agent population, letting the agents interact, and monitoring what happens. That is, executing the model—spinning it forward in time—is all that is necessary in order to “solve” it.³ Furthermore, when a particular instantiation of an agent-based model, call it A , produces result R , one has established a sufficiency theorem, that is, the formal statement R if A [Newell and Simon, 1972: 13].⁴

There are, ostensibly, several *advantages* of agent-based computational modeling over conventional mathematical theorizing. First, as described above, it is easy to limit agent rationality in agent-based computational models. Second, even if one wishes to use completely rational agents, it is a trivial matter to make agents heterogeneous in agent-based models. One simply instantiates a population having some distribution of initial states, e.g., preferences. That is, there is no need to appeal to representative agents. Third, since the model is “solved” merely by

¹ Indeed, in their most extreme form, agent-based computational models will not make any use whatsoever of explicit equations. Thus, there is a definite sense in which the two distinct branches of computational economics are nearly antithetical. Stated differently, about the only thing that they have in common is that both use computers. Note that there is a close analogy between these two types of computational economics and the situation in computational physics, where the numerical solution of continuum (differential equation) models is only weakly related to particle and cellular automata models.

² Epstein and Axtell [1996] give a fairly comprehensive bibliography of agent-based models in the social sciences that were either in working paper form or published by 1996. Since then there has been a rapid expansion of agent-based modeling efforts, and anything like a complete listing of this work would reference several hundred papers.

³ Of course, if the model is stochastic, then multiple realizations are necessary in order to characterize R .

⁴ “To take sufficiency as a first requirement of a theory is simply to adopt a particular approximating sequence in science’s progress. Since not all things can be done first, a particular theoretical orientation gets some of its flavor from what it puts first.” [Newell and Simon, 1972: 13].

executing it, there results an entire dynamical history of the process under study. That is, one need not focus exclusively on the equilibria, should they exist, for the dynamics are an inescapable part of running the agent model. Finally, in most social processes either physical space or social networks matter. These are difficult to account for mathematically except in highly stylized ways. However, in agent-based models it is usually quite easy to have the agent interactions mediated by space or networks or both.

However, the agent-based modeling methodology has one significant *disadvantage* vis-a-vis mathematical modeling. Despite the fact that each run of such a model yields is a sufficiency theorem, a single run does not provide any information on the robustness of such theorems. That is, given that agent model A yields result R , how much change in A is necessary in order for R to no longer obtain? In mathematical economics such questions are often formally resolvable via inspection, simple differentiation, the implicit function theorem, comparative statics, and so on. The only way to treat this problem in agent computing is through multiple runs, systematically varying initial conditions or parameters in order to assess the robustness of results. While the curse of dimensionality places a practical upper bound on the size of the parameter space that can be checked for robustness, it is also the case that vast performance increases of computer hardware are rapidly converting what was once perhaps a fatal difficulty into a manageable one.

Agent-Based Computation: Architecture and Implementation

Before moving on to discuss distinct motivations for agent-based models, it will serve as useful background to first describe their basic computational structure. Of course, agents are the key ingredient. Each agent possesses both states (i.e., data, also known as instance variables) and rules of behavior (i.e., procedures or functions, aka methods) and are most conveniently represented in software as objects. Agent states can be either private or public; the latter are visible to other agents while the former are not. Similarly with agent behavioral rules — some are private and some public. This is exemplified by the following pseudo-code:

```

Agent object:
  private states:
    preferences;
    wealth_1;          /*      private wealth      */
    :
    ;
  public states:
    bid-price;
    wealth_2;         /*      public wealth      */
    :
    ;
  private behavior:
    compare_choices;
    compute_internal_valuations;
    draw;
    :
    ;
  public behavior:
    initialize;
    seek_trade_partner;
    communicate_with(Agent i)
    :
    ;
end.

```

Code fragment 1: Typical agent object.

Just as each agent is an object, so with the agent population as a whole. It has both states — data relating to the agents — and functions — such as routines to compute population statistics. An example population object is:

```

AgentPopulation object:
  private states:
    internal_representation_of_population;
    currently_active_agent;
    :
    ;
  public states:
    number_of_agents;
    number_of_agents_working_in_firms;
    :
    ;
  private functions:
    get_Nth_agent (N);
    randomize_the_agents;
    draw;
    :
    ;
  public functions:
    initialize;
    agents_trade;
    compute_average_bid_price(commodity j);
    :
    ;
end.

```

Code fragment 2: Typical population object.

In order to track the performance of an agent model some statistical analysis must be hooked into the code. This can take the form of simple text output, graphical display or real-time, model-generated statistics, or even real-time econometric estimation. Then with all this in place an agent-based computational model becomes little more than:

```

program typical_agent_model;
  initialize_agents;
  repeat:
    agents_interact;
    compute_statistics;
  until done;
end.

```

Code fragment 3: Typical agent-oriented program.

Much detail has been abstracted away in these code fragments. In actual implementations agent interactions occur either sequentially or in parallel, and if in parallel then with some degree of synchrony. The timing of interactions implicitly specifies a global clock and the definition of a model “period.” The order of agent activation must be systematically randomized from period to period in order to avoid the production of artifacts, phenomena that arise due to accidentally imposed inter-agent correlations and that are not robust to seemingly innocuous code changes. It is sometimes necessary to give each agent its own random number generator. And so on. But these code fragments give a skeletal picture of agent-model architecture.

There is a definite sense in which agent-based computational models are relatively *easy* to create, in comparison to other computational models. This is so because the heart of the code are the agent behavioral methods, and you write these only once; the behavioral repertoire is the same for each agent. Consider the following example. An agent-based model of exchange is instantiated with 10^6 agents. Each agent has preferences, an endowment, and a current allocation. Depending

on the size of the commodity space, each agent might consume $O(10)$ to $O(1000)$ bytes. Overall, the amount of memory used by the agent-model is $O(10^7)$ to $O(10^9)$ bytes, that is from 10 megabytes to perhaps a gigabyte. But such a program can be specified in fewer than 1000 lines of C/C++ code, perhaps 100 lines of SWARM or Ascape.⁵ So a relatively short “program” at compile-time is actually a very large “program” at runtime.⁶

This architecture, in which very little source code effectively controls a much larger amount of execution code, is the basis for the highly scalable nature of agent-based models. The number of agents or commodities, for instance, are conveniently specified as user-defined constants in the source code or read in from the command line, and thus the scale of the model can be specified at compile or run-time. Typically, no significant rewriting of the code is needed in order to change the scale of the model.⁷

It is also the case that the “small source, large execution code” character of agent computing is partially responsible for the production of artifacts, an important class of systematic problems that can arise in agent models, as alluded to above. When a small amount of code — say a single commodity exchange rule, for example — controls so much other code, then it will sometimes be the case that an idiosyncrasy in the rule code will produce output that one might erroneously take to be a significant result of the model. A common route to phenomena of this type occurs when the agent interaction methods impose some spurious correlation structure on the overall population — for instance, agents are interacting with their neighbors more than with the population overall in what is supposed to be a “soup” or “mean field” interaction model — then an ostensibly systematic result — large price variance, say — is clearly artifactual.⁸ There is no real solution to this problem, aside from careful programming. One can, however, look for the existence of such artifacts by making many distinct realizations of an agent model, perturbing parameters and rules. When small perturbations in the code produce large changes in the model output, then artifacts may be present. Sometimes, large changes in output are realistic and not signatures of artifacts. For example, imagine that a small change to a threshold parameter makes an agent die earlier than it otherwise would, and therefore induces at first a small change in agent exchange histories (i.e., who trades with who), that over time is magnified into a wholesale change in the networks of agent exchange. Perhaps this is not unrealistic. But when such large scale changes have origins that are unrealistic empirically, then one should be instantly on guard for undiscovered flaws in the source code.

In the next section we describe a simple use of agent-based models and argue that the term “simulation” is best applied in such cases. Then, in section 3 certain commonly encountered difficulties with mathematical models are described, as is the use of agent-based models to circumvent these problems. Finally, in section 4 a third use of agent-based models is described in relation to problems for which mathematical representation is simply not useful.

⁵ For more on SWARM, a high-level language for agent-based modeling, see Minar *et al.* [1996] and Daniels [2000]. For a description of Ascape, see Parker [2000].

⁶ The standard notion of “program” seems problematical in this context. A large-scale agent model instance closely resembles a dynamic database, in which the agent object population represents the data and the agent interaction specifications are responsible for the dynamics.

⁷ Of course, this is not the same as saying that the number of computations (clock cycles, agent-agent interactions) is a linear function of the scale of the model. The computational complexity of such models is a much more complicated issue and will be addressed below.

⁸ For a discussion of such artifacts in a particular case see Huberman and Glance [1993].

FIRST USE — WHEN MODELS CAN BE FORMULATED AND COMPLETELY SOLVED: AGENT MODELS AS CLASSICAL SIMULATION

Here we describe a simple, perhaps some will say trivial, use of agent-based computational modeling. It is almost certainly *not* the most important use of agents, but it is the use that best fits the conventional meaning of “simulation.”

Imagine that some social process is under study, and that it is possible to write down one or more mathematical relationships that fully describe the process. Furthermore, imagine that the resulting model may be solved explicitly, either symbolically or numerically. Then what role is there for an agent-based computational model of the process?

There are many answers to this question. First, if the solution is not available symbolically, but only numerically, then an agent-based model serves as a useful check on the numerical solution. That is, the solution obtained by numerically solving the equations should agree with that which emerges from the agent model. As an example of this, consider solving some set of equations for an income distribution. Alternatively, one creates an agent-based model of the process, runs the model, queries each agent for its income, sorts the data, and builds-up an ‘empirical’ income distribution, which should agree with the numerically-solved equations. Because these methods are equivalent, it would seem that researchers rarely build both numerical and agent-based models.⁹ In fact, this is not a common use of agent-based computational models.

Agent Computational Modeling as a Type of Monte Carlo Simulation

A second role for agent computing occurs when a model is stochastic, in which case the solution will be some distribution of outcomes. This is classical simulation as practiced in operations research, a well-known variety of which is Monte Carlo simulation.¹⁰ In particular, imagine that the output, Y , of a stochastic model is given by $f(X)$, where f is a deterministic function of a random variable, X . With X and f given, Y is completely determined although often cannot be computed analytically. Thus one resorts to Monte Carlo analysis, in which many realizations of $X = x$ are made and for each one $y = f(x)$ is computed. In this way Y is built up progressively.

There is a one-to-one relationship between this kind of Monte Carlo analysis and agent-based modeling. In particular, if X is a known distribution over a heterogeneous agent population — e.g., agent preferences — and f is some specified social process — yielding equilibrium allocations of goods, say — then each realization of X can be thought of as an agent. Therefore, it is only this narrow usage of agent-based modeling — when the model is intrinsically stochastic and the equations governing it cannot be solved analytically — that deserves to be called *simulation*, or so it seems to me. An example attempts to flesh out this usage of agent-based models as simulation.

⁹ Indeed, the only example of this that I am aware of is Carpenter [forthcoming].

¹⁰ See, for instance, Bratley *et al.* [1987] or Banks and Carson [1984].

Example 1: Consider the classical OR simulation of a bank teller line. This is a queuing model and no general analytical solution is known for arbitrary distributions of arrivals and service times. Therefore, the queuing process is commonly simulated via the Monte Carlo method and distributions of waiting times and server utilization result. However, this is completely equivalent to actually instantiating a population of agents, giving them heterogeneous arrival times according to some distribution, and then running the agent-based and studying the queue lengths that emerge, over time building up the entire waiting time distribution function.

So, whenever stochastic governing equations of a social process can be written out and their solution space characterized, so that all that remains to be done is generate numerical realizations, then the use of the term *simulation* to describe agent-based computational models corresponds to its traditional usage in operations research.

The Efficacy of Agent Computing as a Tool for Presenting Mathematical Results

Yet another role for agent-based computing occurs in the (increasingly unusual) case in which a model of a social process can be solved explicitly. Here it would seem that there is no role whatsoever for agent computing, since the solution is completely specified. However, even in this case there is utility for creating an agent-based implementation of the formal model. Most people outside academia have modest knowledge of mathematics and statistics, and therefore have a difficult time interpreting numerical model output, or understanding regression results, for example. But people are very good at pattern recognition and analogical reasoning. Because the output of agent models tends to be visual, such models can be very effective at depicting formal results from mathematical models. Such uses of agents are especially relevant for demonstrating technical results to policy-makers and business decision-makers.

SECOND USE — PARTIALLY SOLUBLE MODELS: ARTIFICIAL AGENTS AS COMPLEMENTARY TO MATHEMATICAL THEORIZING¹¹

The second distinct use of agent-based computational models occurs when it is not possible to completely solve a mathematical model analytically. Here, theory yields mathematical relationships, perhaps even equations, but these are not directly soluble, or perhaps no appropriate solution concept is readily available. Or it may be that an equilibrium configuration can be figured out, but that its stability is ambiguous. Or perhaps the dependence of the equilibrium on a distinguished parameter is of interest but cannot be readily computed. There are a variety of ways in which formal models resist full analysis. Indeed, it is seemingly only in *very* restrictive circumstances that one ever has a model that is completely soluble, in the sense that everything of importance about it can be obtained solely from analytical manipulations.

In such circumstances it is common to resort to numerical instantiations of the symbolic model, in order to glean some additional understanding. It is also generally possible to build agent-based computational models in order to gain insight into the functioning of the model. Now, if the agent-based computational model is merely an instantiation of the mathematical model, then we are back to agents-as-simulation, described in the previous section. However, in the process of formalizing a theory into mathematics it is often the case that one or more —

¹¹ For a related viewpoint in the context of non-agent computational models, see Judd [1997].

usually many! — assumptions are made for purposes of simplification; representative agents are introduced, or a single price vector is assumed to obtain in the entire economy, or preferences are considered fixed, or the payoff structure is exactly symmetrical, or common knowledge is postulated to exist, and so on. It is rarely *desirable* to introduce such assumptions, since they are not realistic and their effects on the results are unknown *a priori*, but it is expedient to do so.

As described in the first section of this paper, it is typically a relatively easy matter to relax such ‘heroic’ assumptions-of-simplification in agent-based computational models: agents can be made diverse and heterogeneous prices can emerge, payoffs may be noisy and all information can be local. One can then “dock” the agent-based computational model with what may now seem like the highly stylized analytical model by creating an instance of the computational model with homogeneous agents, fixed preferences, etc. Then, once the “docked” computational model is shown to reproduce the known analytical results — thus providing a crude, first-order validation of its performance — it can be instantiated with fully heterogeneous agents, etc., and then used for systematic study.¹² In this usage agent-based computational modeling turns out to be very powerful at advancing one’s understanding of a formal theory. It is as if the agent model is a “prosthesis for the imagination” and a complement to formal theory.

An argument sometimes used against agent-based modeling is that individual realizations (runs) are just special cases and nothing very general can really be known about the process under study until analytical results are obtained. We have offered the ‘computer programs as sufficiency theorems’ argument of Newell and Simon, in the introduction to this paper, as a partial refutation of this criticism, but more can be said. One role for agent computing is to check whether mathematical results still hold when specific assumptions are relaxed. In fact, if a single instance of an agent-based model produces results that violate a theorem that holds in more restricted circumstances, then the computational model stands as a counter-example to the wider applicability of the theorem. This use of agent computing, while usually not explicit, is implicit in much of the work that has appeared to date.

In this section, usage of agent-based computational models as complementary to mathematical social theory will be described. Formal models for which analytical results are incomplete will be stated. Then the ways in which agent-based computational models can be used to further explore the solution space will be presented.

Equilibria Exist But Are Effectively Uncomputable

Fixed-point theorems were apparently introduced into economic theory by von Neuman in his input-output model [1945–46]. Now, many domains of economic theory depend on fixed-point theorems for the existence of equilibria. The original Brouwer and Kakutani theorems did not have constructive proofs, but such proofs are now known, e.g., through Sperner’s lemma. Of course, the existence of equilibrium is not the same as its achievement. That is, without some mechanism for converging to a fixed point in a bounded amount of time then there is little reason to believe that a fixed point could ever be realized; i.e., equilibrium would not be plausible.

One is tempted to conceive of economic processes literally in terms of the algorithm used in constructive proofs of fixed point theorems. But this research program is doomed to failure since it is known that the Brouwer theorem has worst case computational complexity that is

¹² For more on “docking” models, see Axtell, Axelrod, Epstein, and Cohen [1996].

exponential in the dimension of the problem [Hirsch *et al.*, 1989] — the dimension being the size of the commodity space in the Arrow-Debreu version of general economic equilibrium. Furthermore, it has recently been shown that computation of Brouwer and Kakatani fixed points are computationally hard problems in general [Papadimitriou 1994]. Taken together with Scarf's empirical estimate [Scarf 1973] that the number of computations required to equilibrate a computable general equilibrium (CGE) model scales like the size of the commodity space to the fourth power, one is left to believe that the Walrasian model, in which no trade occurs until a benevolent auctioneer first computes the market-clearing price vector, is not a particularly credible picture of how a real exchange economy works.

Example 2: An agent-based computational model of exchange has been created and its computational complexity investigated [Axtell 2000b]. In this model agents from a heterogeneous population are paired at random and engage in Pareto improving bilateral exchange. It has been demonstrated that the number of agent-agent interactions required to equilibrate the economy is linear in the number of agents and quadratic in the number of commodities. These results are robust over five decades of agent population size (10 to 10^6 agents), and nearly four decades of commodity space dimension (2 to 10^4 goods). Now, it turns out that it is possible to prove some theorems for this process — that it converges, that individual wealth can change over time, that it is possible to obtain Walrasian equilibria via bilateral exchange but bilateral exchange equilibria cannot be achieved by a Walrasian mechanism. The development of these results was stimulated directly to studying the output from the agent-based model.

As a final point about the existence of fixed points, it would seem useful to point out that in the bilateral trade model described above, there exist unimaginably vast numbers of equilibria.¹³ Existence of equilibria is a relatively trivial matter in these models and the important question becomes equilibrium selection. The equilibria that actually obtain in these agent-based models are always idiosyncratic with respect to the agents, in the sense that from realization to realization any particular agent's allocation may vary substantially. But the macrostatistics of these models are quite robust across realizations. Thus we have a weak form of *path dependence* in which the history of agent interactions is important for the individuals, although not for the economy overall. Taking these models seriously leaves one with the impression that sociology — in particular, social networks determining who interacts with whom — should be given a more prominent place in economic models, while fixed point theorems should be concomitantly demoted.

Equilibrium Not Attained by Boundedly Rational Agents

There are many ways to model bounded rationality.¹⁴ Papadimitriou [1993] has argued that computational complexity is a useful framework for thinking about rationality: full rationality implies the ability to do exponentially difficult computations, while bounded rationality means that the requisite computations are bounded by a polynomial in the relevant problem parameters. There have appeared a variety of models in which bounded rationality is treated in this way, including Board [1994] and Spear [1989]. The main results of these mathematical models is that

¹³ This is also true in the model of coalitions of DeVany [1994] and the computational Tiebout model of Kollman *et al.* [1997].

¹⁴ See the recent review article by Conlisk [1996].

boundedly rational agents have a difficult time of, for example, learning rational expectations equilibria in non-trivial economic environments.

A related way of modeling boundedly rational agents is through the use of finite automata. In game theoretic settings this approach was first adopted by Neyman [1985] and Rubinstein [1986]. There is a large and growing literature on this subject [cf. Binmore and Samuelson [1992]. However, it has recently been shown, in the context of two person games, that the process of learning the strategy of an automaton opponent generally requires an amount of time exponential in the number of states of the automaton [Mor *et al.* 1996; see also Prasad 1997].

Bounded rationality is a common feature of agents in agent-based computational models. The failure to achieve equilibrium is a characteristic result of such models. However, while it is often quite difficult to characterize non-equilibrium phenomena analytically, the ability to systematically study dynamics is one of the powerful features of agent-based computational models, as exemplified by the following.

Example 3: The Santa Fe artificial stock market is an agent-based model in which heterogeneous agents allocate their assets between a risky stock paying a stochastic dividend and a risk-free bond [Arthur *et al.* 1996]. The agents formulate their expectations adaptively, based on past market performance, and these are therefore endogenous to the market. What emerges is an ecology of beliefs among the agents that coevolves with time. There exists a regime in the model in which something akin to rational expectations equilibria obtain, a regime characterized by limited exploration of alternative expectations models by the agents. However, when agents are actively engaged in exploring the space of models then the market self-organizes into a more complex structure, prices and volumes have statistical features more characteristic of actual market behavior, 'technical trading' arises in the agent population, and the market can experience bubbles.

It would seem that computational agent models have much to contribute to the study of bounded rationality.

Equilibria Obtained Asymptotically But Not Realized over Long Periods

In theoretical models, especially stochastic ones, it is common to characterize equilibria asymptotically. The empirical significance of such results hinges on the usually unjustified assumption that the process does not get stuck away from its asymptotic state for significant periods. For if there were long-lived transient states then the empirical data might be described by these, and not the asymptotic state. Consider ergodic models, in which all states are visited with positive probability. Asymptotic states for such systems are states in which the system spends *most* of its time. However, if such a system is confined to a region of the state space for periods of time that are long compared to observation times then the system is said to display 'broken ergodicity.' Agent-based models are effective tools for explicating the structure of such long-lived transient states, as exemplified by the following.

Example 4: Young [1993a, 1993b] has described a class of evolutionary models in which a population of agents repeatedly interacts in the context of a bargaining game. Each agent has finite memory and plays a best reply strategy, based on its idiosyncratic memory. Agents play randomly with a small probability. He has shown that of the many Nash equilibria in such games, only some have positive probability asymptotically, the *stochastically stable equilibria*.

An agent-based computational version of this model has been created [Axtell *et al.*, forthcoming] that (1) visually displays the path of a population to Nash equilibrium configurations, (2) illustrates transits between equilibria, while (3) demonstrating the possibility of achieving the stochastically stable equilibria eventually. Furthermore, the way in which these results depend on the noise level (i.e., probability of playing randomly) has been investigated using this model. This agent-based code is a variant implementation of classical simulation of the Markov process that underlies this game model. Instead of generating random state vectors — i.e., agent memories — and iterating forward to a stochastically stable state, each agent can be thought of as a state vector, and the interactions of the agents as the way in which initially random state vectors get transformed into stochastically stable states. There is even a sense in which the agent-based model is a relatively efficient simulation technique, for classical simulation of the actual governing equations requires storage of a very large, albeit sparse, transition matrix. This agent-based computational implementation also illustrates the broken ergodic character of this model, insofar as the transition times between certain long-lived but not asymptotically-stable states in the model depends exponentially on the agent memory length, population size, and the inverse of the noise level. That is, the population can get stuck away from the stochastically stable equilibria for very long periods of time for large agent memories, large populations or low noise levels. These phenomena are characterized quantitatively by computational model, something very difficult to do analytically, giving further evidence of the fertile complementarity between analytical modeling and agent computing.

Overall, agent computation is particularly well-suited for depicting the transient, far-from-equilibrium behavior of social systems, a point further elaborated in the next section.

Equilibria Exist But Are Unstable

The stability of equilibria can often be inferred from the structure of a problem, either by qualitative considerations or explicit calculation. A problem may have multiple equilibria, not all of which are stable, and sorting the stable from the unstable becomes an important question. What happens when all equilibria are unstable? One's first impulse is to reject such models out-of-hand, thinking that no social process would be well represented by such a model. However, the following example indicates that models with unstable equilibria may be relevant to some social institutions, and that agent-based computational models may be the most efficacious route to developing a thorough understanding of such models.

Example 5: Canning [1995] has analyzed a simple model for the formation of groups. In it, agents must choose how much effort to put into their group, and there is an associated cost to the agent of her effort. There are local economies of scale such that the effective effort of two agents working together in the same group is greater than the sum of their individual efforts; this leads to teams. Each member of a group shares the total output equally; this leads to free riding behavior as the group size grows. That is, utility maximizing agents find it in their self-interest to put in large effort levels when they are receiving direct compensation for their labors, and little effort when their share of output is insensitive to how hard they work. It is possible to analyze this model as a dynamical system. This reveals that the equilibrium group size and effort level are unstable: groups are constantly growing and shrinking — all groups

are meta-stable — the exact nature of the dynamics depending on parameters of the model.¹⁵ It appears that it is not possible to say much more about this model analytically, but important questions remain, for instance, “What is the distribution of group sizes that arises in the model?” But how might one go about analyzing the out-of-equilibrium structure of such a model?

An agent-based computational model of the endogenous group formation process has been created [Axtell 1999]. It displays the essential instability of the mathematical model insofar as firms are born, grow, and then succumb to the free rider problem and either shrink significantly or vanish altogether. It yields a skewed distribution of firm sizes that closely resembles the empirical firm size distribution. The agent model also displays quantitatively correct firm growth rate distributions, dependence of growth rate variance on firm size, and wage-size effects — workers in larger firms are paid more. Here the agent-based served to connect the agent (i.e., microeconomic) specification to aggregate empirical data.

There is an analogy to be made here. In fluid mechanics it is possible, for certain geometries, to solve a set of partial differential equations for the steady-state distribution of fluid velocity as a function of spatial coordinates — the so-called laminar velocity profile. However, even undergraduates are taught that this laminar solution to the equations is unstable beyond a certain critical value of a dimensionless ratio of parameters known as the Reynolds number. Of course, just because the steady solution is unstable does not mean that the fluid does not flow! For super-critical Reynolds numbers the flow becomes turbulent, it is unsteady (time-dependent) microscopically, although perhaps stationary, and no general solution to the governing equations is known. If one wants to learn about the structure of turbulent flows, then computational models are essentially the only available methodology.¹⁶ Such models well describe how the qualitative structure of the flow changes with increasing Reynolds number, from essentially laminar flow with small eddies to full-blown turbulence with eddies on all scales. These coherent, self-organized albeit transient structures yield turbulent flow regimes that expend less energy per unit volume than comparable laminar flows, i.e., flows having the same average velocity. Nature is thus not shy about involving non-equilibrium configurations when she finds it appropriate to do so, and economists should be equally intrepid!

Dependence on Assumptions and/or Parameters Unknown

When mathematical models are only incompletely soluble it may prove difficult to determine how the known results depend on particular assumptions or exogenous parameters. In such cases agent-based computational models may prove insightful, as illustrated by the following.

Example 6: Axelrod [1997] describes an agent-based model for the evolution of generic cultural attributes. In the model, agents are arranged in a fixed position on a two-dimensional square grid having N cells on a side, thus N^2 cells total. Each agent is endowed with a set of cultural “tags.” Agents are activated at random to interact with their neighbors, with the

¹⁵ Because the equilibria are unstable does not mean that the model is unrealistic. In actual economies new firms start-up and old firms die every day, a feature difficult to capture with static equilibrium models.

¹⁶ As an aside we note that in computational fluid mechanics the term *simulation* is used to describe model instances and not the general field. Similar usage holds broadly in computational physics.

probability of interaction being proportional to the extent to which the agents are similar.¹⁷ Regions of cultural homogeneity arise in the model, while from region to region cultures are heterogeneous. Once all agents are, with respect to all neighbors, either completely identical — share the same string of cultural attributes — or completely different — no attributes in common — then all cultural evolution has effectively stopped.¹⁸ The number of distinct cultural regions, call it R , is an important summary statistic generated by the model. Axelrod argues that the model is relevant to the origin of distinct languages and language dialects.

Axelrod systematically studied the dependence of R on, among other things, parameter N . Interestingly, he finds that there is a non-monotone relationship between R and N . In particular, for N either small or large, R is small, while for an intermediate value of N there exists a maximum value for R . This is shown in the following figure.

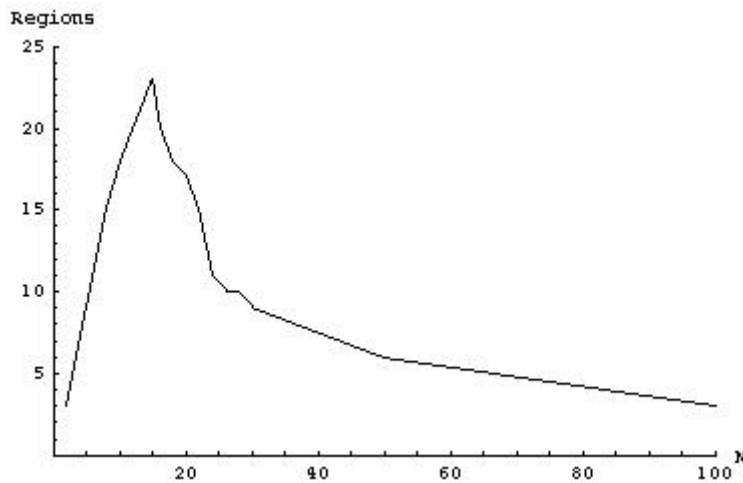


FIGURE 1 Dependence of the equilibrium number of culturally distinct regions on the size of the lattice.

An explanation for this relationship is offered in Axelrod [1997]. In order to assess the reasonableness of this explanation it would be useful have an analytical model of this agent-based model. Axelrod solicited the help of an eminent mathematician in order to formulate this model mathematically, but they were unsuccessful.¹⁹ Therefore, the so-called “docking” experiment described in Axtell *et al.* [1996] was undertaken in order to verify this result.

While the illustration above describes how agent models can be used to understand parameter dependence, their use in assessing the importance of assumptions is directly analogous. For example, if a theorem is proved in a mathematical model under the assumption that all agents have the same number of memory states, one can simply give the agent population in an agent-based model some distribution of memory states, then run the program and observe whether or

¹⁷ This cultural transmission model generalizes that described in Epstein and Axtell [1996]. Some details of this are described in Axtell *et al.* [1996].

¹⁸ There is no mutation in the basic version of the model. For some variations see Axtell *et al.* [1996].

¹⁹ Personal communication.

not the theorem still holds. If it does not then the agent-based model represents a counter-example to generalizing of the theorem with respect to heterogeneous agent memory length.

Relatedly, it is quite common to attempt an assessment of an analytical model's dependence on an assumption, to determine whether or not it is a "key" assumption. If several assumptions are made in a model, it may even be that the importance of each is assessed, and one is heartened if each seems to matter only a little, if at all. However, these assessments are usually made independently — that is, it is unusual to try to relax multiple assumptions at once. But the overall validity of a result may depend on such simultaneous relaxation. Often, agent computational models facilitate model experimentation of just this type.

Equilibrium Configuration Known Only for Simple Interaction Networks

The network aspect of social processes is a topic receiving renewed theoretical interest recently. The models that have appeared have a variety of forms, from random graph models to partial differential equation approaches, e.g., in the 'new economic geography' [Krugman 1996]. Typically, these models are analytically soluble only under rather restrictive conditions, for example, requiring homogeneous agents or limited spatial dimension (e.g., 1D).

Spatial networks are quite naturally represented in agent-based computational models. A physical location can be part of an agent's internal states. Likewise, its position in a social network can be easily represented internally, perhaps through a list in which each agent keeps so-called pointers, or pointers to pointers (handles), to other agents. Then, agents can be made to interact either through the physical space or the social network or both.

Social network interactions have been studied theoretically [cf. Durlauf 1997, Young 1998], but typically these results are only applicable to very specific networks, typically lattices or random graphs. Human social networks seem to have a "small world" property, i.e., they are characterized by both highly "local" and certain long-range connections [cf. Watts 1999]. Today little is known about social processes on such graphs, except that in moving from one graph to another the effects can be significant [Axtell 2000a].

Example 7: Glaeser, Sacerdote and Scheinkman [1996] build and estimate a simple model of criminality based on local social interactions. Large spatial variation is characteristic of crime rate data, and these authors demonstrate that such high variance can be produced by a simple model in which a large proportion of agents imitate their neighbors. They analyze their model mathematically and are able to develop closed form expressions for excess variance across space. Their estimation results support the hypothesis that certain kinds of crime are more social — in their model, require more imitation — than others. Auto theft, for example, is more social than murder. They even obtain crude approximations for the size of criminal groups by crime type.

They analyze the model for agents arranged along one dimension, and in footnotes offer that they have studied a computational model in two and three dimensions. Now, while it is certainly true that humans live in a three-dimensional physical world, human imitative behavior occurs within social networks. It would be very interesting to know how their results stand up when agents behave as they postulate but within realistic social networks. This would seem to be a pregnant domain for agent computing. Furthermore, their behavioral model of the agents is so primitive — some agents are criminal for life, for example — that it would also be

interesting to know how robust their results are to more realistic agent specifications. Once again, this is something easily manageable with artificial agents.

Lastly, for many models the effects of adding spatial dimensions so complicates the analysis that little progress is possible analytically, and resort to computational approaches seems to be the only way forward. The well-known Schelling model (see Epstein and Axtell [1996] and references therein) is an example of this and has been fruitfully studied and extended by various researchers using agents (for instance, Zhang [2000]).

Equilibrium Less Important Than Fluctuations and Extreme Events

In many stochastic dynamic models it is possible to characterize the equilibria and stability of the models asymptotically, but little can be said about their out-of-equilibrium behavior. Transit times between equilibria, expected time spent at particular equilibria, dependence of these times on the level and character of noise, these are all dynamical issues that go beyond mere existence of equilibria.

The importance of the out-of-equilibrium behavior of such systems is clear, since it may take such systems very long times to reach asymptotic equilibrium. In fact, it may be that one only cares about the extreme events of such systems. Agent-based models are good devices for systematically studying such dynamics. The following example illustrates this.

Example 8: In models of traffic, important output statistics are the distributions of jams by size and lifetime. Agent-based models have been created to study the dynamic aspects of traffic [e.g., Nagel and Rasmussen, 1993]. These models are capable of reproducing real-world data with high fidelity [Casti 1997]. In particular, on crowded roads it is known that local flowrate data can be highly non-stationary. Differential equation (fluid mechanical) models of traffic have a difficult time of capturing this feature of the data, as well as faithfully representing other transient and dynamical properties of real world traffic. However, these phenomena do emerge in large-scale, massively parallel (i.e., agent) computational models of traffic. Additionally, the jamming distributions that arise in these models display a kind of universality also seen in statistical physics. That is, the macrostatistics of the systems are relatively insensitive to the agent specifications, i.e., many reasonable models of driving behavior produce the same distribution of traffic jams!

All, Or Nothing at All

A characteristic feature of agent computing, although not often noted, is that once a model has been created it provides not merely one aspect of the solution — the equilibria, say, or the stability — but rather entire solution trajectories. As a functioning agent model spins forward in time it gives first the out-of-equilibrium dynamics, then possibly approaches an equilibrium configuration and either becomes entrained in some basin of attraction or not, thereby indicating something of the stability of the equilibria. Repeated realizations yield the dependence of results on parameters and the importance of assumptions — all this from a single operational piece of code. Of course, if the code is not working properly or generates artifacts then nothing useful comes out at all.

THIRD USE — MODELS OSTENSIBLY INTRACTABLE OR PROVABLY INSOLUBLE: AGENT COMPUTING AS A *SUBSTITUTE* FOR ANALYSIS

When a model of a social process is completely intractable, either apparently or provably so, then there would seem to be little hope of progress with purely symbolic manipulations. We investigate these two cases of intractability in turn.

Intractable Models

Claims concerning a model's intractability are not often made — at least not in print — perhaps because people do not like to sound like they have given up. If one only had more time to read through the topology book or the new probability journal, perhaps a breakthrough could be had. Or maybe the problem can be 'saved' for a bright student. These are admissions that a model is in some way intractable, call it 'locally intractable.'

For example, it is well-known that there do not exist closed form solutions to certain relatively simple differential equations in terms of elementary functions. When a problem is intractable in this way it has nothing to do with its complexity. Rather, it is an artifact of the limited explorations undertaken to date in the infinite library of functions. In such circumstances one makes recourse to numerical solution. But there are also instances in which numerical solution would appear to be essentially intractable, not in the sense of being impossible but merely not useful. This occurs when governing equations are highly nonlinear.²⁰ When such circumstances arise in computational physics, particle models can sometimes be advantageous. We believe the same is true of agent-based computational models in the social sciences: if it is hard to make *any* real progress solely by analytical manipulation, agent models may prove useful.

Model Formally Undecidable

²⁰ According to Feigenbaum [1988: 567]:

Consider a cloudlike initial configuration of some fluid equation (a classical field theory). Imagine that the density of this configuration possesses rich scaling properties (e.g., a fixed spatial scale exponent over many decades). Moreover, imagine that at successive moments of time it also possesses these scaling properties, although possibly variable in time. From this we should surmise that the instantaneous velocities should also possess similar scaling properties. Imagine that these scalings are easily specified, that is, we have discerned in this complex spatial object some prescriptive rules that if iterated would construct it. Now let us contemplate how we advance this structure in time. By the locality of the field equations we must *actually* spin out this iterative construction in order to provide the equations with the sort of initial data they require. Now we can advance the structure a step ahead in time. But what do we now have? Simply an immense list of local density and velocity values of high local irregularity. Of course, if we possess a good algorithm, we could now from this new pabulum of data again discern the scaling information — perhaps evolved — that we know about anyway. This is obviously a foolish double regress. Since our informed understanding lay in the scaling description, we should obviously have transcribed our “true” local dynamics into one pertinent to these scalings, rather than mount a numerical program that strains the most powerful machines we possess. That is, the solution in the usual sense of our local field theories is apt to be a mindless enterprise when the solutions happen not to be simple. In this sense, our theories, while “true,” are useful only to God, which seems not to be the hallmark of what humans adjudge to be truth.

A conventional view of Gödel's incompleteness theorem is that it is largely irrelevant to the daily practice of mathematics.²¹ Chaitin [2000] has argued for essentially the opposite perspective: the problems of incompleteness are ubiquitous and place a serious constraint on what can be accomplished with mathematics. In the following example decidability rears its head in a context closely related to a variety of social science problems.

Example 8 [Buss *et al.* 1991]: Consider the following system of coupled automata. There are A identical automata, each having some finite number of states, S . The initial state of each automaton is given; an A -vector specifies all states. There is a global control rule, G , which for simplicity take to be a first-order sentence. Finally, there is some time, T , at which we wish to know the state of the system. Call this system is *predictable* if it is possible to determine its state at T in an amount of time that is polynomial in A , S , and the logarithm of T . Formally, if the system is not predictable in this way, then it is PSPACE-complete, and the only way to effectively predict the state vector at T is to build an automata (multi-agent) model and execute it; that is, realizing an agent-based model is the best one can do.

Definition: A poll, $P(x)$, gives the number of automata in state x .

Definition: G is *constant-free* if it does not refer to any particular state of the automata.

Instance of a constant-free rule: If $\forall x \exists y$ such that $P(x) = P(y)$ then TRUE else FALSE.

Instance of a non-constant-free rule (referendum on s_i): If $P(s_i) > A/2$ then TRUE else FALSE.

Instance of an apparently non-constant-free rule that is actually constant-free: If $\forall x P(x) = P(s_i)$ then TRUE else FALSE; this rule is actually constant-free since it is equivalent to: If $\forall x \forall y P(x) = P(y)$ then TRUE else FALSE.

Theorem 1: If G is constant-free then the system can be predicted in polynomial time.

Theorem 2: If G is not constant-free then the system is PSPACE-complete.

Theorem 3: The set of G that are constant-free is undecidable.

Many aspects of this example have much in common with modern, mathematical social science. Agents are interacting, they are changing state over time while engaged in a kind of voting behavior. Perhaps progress in social science will come from a better understanding of such minimal, abstract formulations of social processes, despite their provable intractability.

CONCLUSIONS

It has been argued that there exist three distinct uses of agent-based computational models. First, when numerical realizations are relevant agents can perform a variant of classical simulation. Second, when a model is only incompletely solved mathematically — its equilibria unknown, stability of equilibria undetermined, or the dependence on parameters opaque — then an agent-based model can be a useful tool of analysis, a complement to mathematics. Third, there

²¹ For a typical statement of this type see the October 1997 *SIAM News* and the review of “Logical Dilemmas: The Life and Work of Kurt Gödel.” The reviewer writes “...it is an important fact that the brilliant, earth-shaking theorems of Gödel are of absolute unimportance to 99.5% of research mathematicians in their professional work.”

are cases in which mathematical models are either apparently intractable or provably insoluble. In such circumstances it would seem that agent computing is perhaps the only technique available for systematic analysis, a substitute for formal mathematical analysis.

There are two other reasons that augur for the increased prominence of agent-based models in the social scientist's toolkit. First and foremost, continued hardware evolution will soon place on our desks machines with capabilities beyond John von Neumann's wildest dreams. These machines will have the capability of modeling 10^6 agents of reasonable complexity, even larger numbers of simpler agents. The positive program of creating entire mini-economies in silicon will be possible, if we can just learn how to build sufficiently realistic agents: agents who trade in markets, who form firms, who procreate, who engage in political activity and write constitutions and bribe other agents for votes while trying to pass term limits. Today we do not know how to do all these things.

The second reason to work with computational agents is that this is a critical time for software agent technology. There has been a sudden flurry of activity around agents, particularly in computer science and information technology, where engineers and business people are busy populating the Internet with tradebots, automated markets, auction engines, and other agent-like devices. But basic foundational issues remain open questions: What agent designs lead to robust system-level performance? Does the distributed, decentralized nature of agent systems make them more or less exploitable than systems having more centralized control?

There is irony in this arrival of agents — in many ways a foreign methodology — to economics and political economy. A key theme in agent computing is that of decentralization, that realistic social processes can be seen to emerge through the interactions of individuals. While decentralization is also an important theme in modern, mathematical social science, so is a competing "social planner" view, in which optimal outcomes are seen to be the result of a benign 'auctioneer,' who has perfect information and infinite computing power at its disposal, who can design perfect mechanisms and who can compute, implement and enforce optimal tax rates. This has given a peculiar methodological orientation to certain sub-fields, especially in economics, of which general equilibrium is an example. But things are changing, as Rust [1996] writes:

"The reason why large scale computable general equilibrium problems are difficult for economists to solve is that they are using the wrong hardware and software. Economists should design their computations to mimic the real economy, using massively parallel computers and decentralized algorithms that allow competitive equilibria to arise as 'emergent computations'...[T]he most promising way for economists to avoid the computational burdens associated with solving realistic large scale general equilibrium models is to adopt an "agent-based" modeling strategy where equilibrium prices and quantities emerge endogeneously from the decentralized interactions of agents."

Today it is conventional practice among mathematical social scientists to state a model in mathematical terms and then attempt to solve it analytically. Only later, if at all, does one resort to a computational — usually numerical — model, having encountered analytical difficulties. However, given that agent computational models are a powerful analytical tool as well as an effective presentation technology, one wonders whether such models might not someday be the first line of attack on new problems, with authors resorting to mathematics only to "tidy up" what the agent model has clearly demonstrated to be a robust feature of the problem. Perhaps as printed journals, with their static equations and figures, give way to electronic, hypertext journals

and dynamic, downloadable model animations, there will come a day when we will all wonder how we ever got along without agents.

ACKNOWLEDGMENTS

A preliminary version of this paper was circulated under the title “Three Distinct Uses of Agent-Based Computational Models in the Social Sciences.” For useful comments I thank Art DeVany, Joshua Epstein, Carol Graham, Fredrik Liljeros, Peyton Young, and conference participants at Chicago and UCLA.

REFERENCES

- Amman, H.M., D.A. Kendrick, and J. Rust, eds. 1996. *Handbook of Computational Economics*. North-Holland: New York.
- Angeline, P., R.G. Reynolds, J.R. McDonnell, and R. Eberhart, eds. 1997. *Evolutionary Programming VI*. Springer-Verlag: New York.
- Arthur, W.B., J.H. Holland, B. LeBaron, R. Palmer, and P. Tayler. 1996. Asset Pricing under Endogenous Expectations in an Artificial Stock Market. Working paper.
- Axelrod, R. 1997. The Dissemination of Culture: A Model with Global Polarization. *Journal of Conflict Resolution*, 41: 203-226. Reprinted as Chapter 7 in Axelrod [1997].
- Axelrod, R. 1997. *The Complexity of Cooperation*. Princeton University Press: Princeton, New Jersey.
- Axtell, R. 1999. The Emergence of Firms in a Population of Agents. Working paper. The Brookings Institution. Available at www.brookings.edu/es/dynamics/papers.
- Axtell, R.L. 2000a. “Effects of Interaction Topology and Agent Activation Regime in Several Multi-Agent Systems.” Working paper. The Brookings Institution. Available at www.brookings.edu/es/dynamics/papers.
- Axtell, R.L. 2000b. The Complexity of Exchange. Working paper. The Brookings Institution. Available at www.brookings.edu/es/dynamics/papers.
- Axtell, R., R. Axelrod, J.M. Epstein, and M.D. Cohen. 1996. Aligning Simulation Models: A Case Study and Results. *Computational and Mathematical Organization Theory*, 1 (2): 123-141. Reprinted as Appendix A in Axelrod [1997].
- Axtell, R., J.M. Epstein, and H.P. Young. Forthcoming. Emergence of Classes in a Multi-Agent Bargaining Model. In S. Durlauf and H.P. Young, eds. *Social Dynamics*. MIT Press: Cambridge, Mass. Also working paper. The Brookings Institution. Available at www.brookings.edu/es/dynamics/papers.
- Banks, J. and J.S. Carson, III. 1984. *Discrete-Event System Simulation*. Prentice-Hall: Englewood Cliffs, New Jersey.

- Binmore, K. and L. Samuelson. 1992. Evolutionary Stability in Repeated Games Played by Finite Automata. *Journal of Economic Theory*, 57: 278-305.
- Board, R. 1994. Polynomially Bounded Rationality. *Journal of Economic Theory*, 63: 246-270.
- Bratley, P., B.L. Fox, and L.E. Schrage. 1987. *A Guide to Simulation*. Second edition. Springer-Verlag: New York.
- Buss, S.R., C.H. Papadimitriou, and J.N. Tsitsiklis. 1991. On the Predictability of Coupled Automata: An Allegory about Chaos. *Complex Systems*, 5: 525-539.
- Canning, D. 1995. Evolution of Group Cooperation through Inter-Group Conflict. Working paper. Department of Economics, Queens University of Belfast, Northern Ireland.
- Carpenter, J.P. Forthcoming. Evolutionary Models of Bargaining: Combining Agent-Based and Discrete Simulations of Convention Evolution. *Evolutionary Economics*.
- Casti, J. 1997. *Would-Be Worlds*. Wiley: New York.
- Chaitin, G. 2000. A Century of Controversy Over the Foundations of Mathematics. *Complexity*, 5, 5: 12-21.
- Chen, S.-H. and C.-H. Yeh. 1997. Modeling Speculators with Genetic Programming. In Angeline *et al.* [1997].
- Conlisk, J. 1996. Why Bounded Rationality? *Journal of Economic Literature*, XXXIV: 669-700.
- Daniels, M. 2000. Integrating Simulation Technologies with Swarm. In C.M. Macal and D. Sallach, eds. *Proceedings of the Workshop on Agent Simulation: Applications, Models and Tools*, Argonne National Laboratory: Argonne, Illinois.
- DeVany, A. 1994. Hard Cores, Soft Cores, and the Emergence of Self-Organized Coalitions. Working paper. UCLA, Center for Computable Economics: Los Angeles, Calif.
- Durlauf, S.N. 1997. Statistical Mechanics Approaches to Socioeconomic Behavior. Arthur, W.B., S.N. Durlauf and D.A. Lane, eds. *The Economy as an Evolving Complex System II*. Santa Fe Institute Studies in the Sciences of Complexity. Proceedings Volume XXVII. Addison Wesley: Reading, Mass.
- Epstein, J.M. and R. Axtell. 1996. *Growing Artificial Societies: Social Science from the Bottom Up*. Brookings Press and MIT Press: Washington, D.C., and Cambridge, Mass.
- Feigenbaum, M. Presentation Functions, Fixed Points, and a Theory of Scaling Function Dynamics. *Journal of Statistical Physics*, 52 (3/4): 527-569.
- Gilli, M., ed. 1996. *Computational Economic Systems: Models, Methods & Econometrics*. Kluwer: New York.

- Glaeser, E., B. Sacerdote, and J. Scheinkman. 1996. Crime and Social Interactions. *Quarterly Journal of Economics*: 507-548.
- Hirsch, M.D., C.H. Papadimitriou, and S.A. Vavasis. 1989. Exponential Lower Bounds for Finding Brouwer Fixed Points, *Journal of Complexity*, 5: 379-416.
- Huberman, B.A. and N.S. Glance. 1993. Evolutionary Games and Computer Simulations. *Proceedings of the National Academy of Sciences, USA*, 90 (August): 7716-7718.
- Judd, K. 1997. Computational Economics and Economic Theory: Substitutes or Compliments? *Journal of Economic Dynamics and Control*, 21 (6): 907-942.
- Kollman, K., J.H. Miller, and S.E. Page. 1997. Political Institutions and Sorting in a Tiebout Model, *American Economic Review*, 87 (5): 977-992.
- Krugman, P. 1996. *The Self-Organizing Economy*. Blackwell: Cambridge, Mass.
- Minar, N., R. Burkhart, C. Langton, and M. Askenazi. 1996. The SWARM Simulation System: A Toolkit for Building Multi-Agent Simulations. Working paper. Santa Fe Institute: Santa Fe, New Mexico.
- Mor, Y., C.V. Goldman, and J.S. Rosenchein. 1996. Learn Your Opponent's Strategy (in Polynomial Time)! In G. Weiss and S. Sen, eds. *Adaptation and Learning in Multi-Agent Systems*. Springer-Verlag: New York.
- Nagel, K. and S. Rasmussen. 1994. Traffic on the Edge of Chaos. Working Paper 94-06-032, Santa Fe, Santa Fe Institute: N.M.
- Neumann, J. von. 1945-46. A Model of General Economic Equilibrium. *Review of Economic Studies*, 13: 1-9.
- Newell, A. and H.A. Simon. 1972. *Human Problem Solving*. Prentice-Hall: New York.
- Neyman, A. Bounded Complexity Justifies Cooperation in Finitely Repeated Prisoner's Dilemma. *Economics Letters*, 19: 227-229.
- Papadimitriou, C.H. 1994. On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence. *Journal of Computer and System Sciences*, 48: 498-532.
- Papadimitriou, C. 1993. Computational Complexity as Bounded Rationality. *Mathematics of Operations Research*.
- Parker, M. 2000. Ascape: An Agent-Based Modeling Framework in Java. In C.M. Macal and D. Sallach, eds. *Proceedings of the Workshop on Agent Simulation: Applications, Models and Tools*, Argonne National Laboratory: Argonne, Illinois.
- Prasad, K. 1997. On the Computability of Nash Equilibria. *Journal of Economic Dynamics and Control*, 21 (6): 943-953.

- Rubinstein, A. 1986. Finite Automata Play the Repeated Prisoners' Dilemma. *Journal of Economic Theory*, 39: 83-96.
- Rust, J. 1996. Dealing with the Complexity of Economic Calculations. Working paper. Department of Economics, Yale University: New Haven, Connecticut.
- Simon, H. 1996. *The Sciences of the Artificial*. Third edition. MIT Press: Cambridge, Mass.
- Spear, S.E. 1989. Learning Rational Expectations under Computability Constraints. *Econometrica*, 57: 889-910.
- Watts, D.J. 1999. *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press: Princeton, New Jersey.
- Young, H.P. 1993a. The Evolution of Conventions. *Econometrica*, 61 (1): 57-84.
- Young, H.P. 1993b. An Evolutionary Model of Bargaining. *Journal of Economic Theory*, 59(1): 145-168.
- Young, H.P. 1998. Diffusion in Social Networks. Working paper. Brookings Institution: Washington, D.C. Available at www.brookings.edu/es/dynamics/papers.
- Zhang, J. 2000. Residential Segregation in an All-Integrationist World. Working paper. The Johns Hopkins University: Baltimore, Maryland.

Why Agents? On the varied motivations for agent computing in the social sciences. Robert L. Axtell. On Generating Hypotheses Using Computer Simulations. Kathleen M. Carley. The Computer as a Laboratory.Â Advancing the Art of Simulation in the Social Sciences. Robert Axelrod. Multi-Agent Systems and Role Games: Collective learning processes for ecosystem management.