

Research Article

GridLAB-D: An Agent-Based Simulation Framework for Smart Grids

David P. Chassin,^{1,2} Jason C. Fuller,¹ and Ned Djilali^{2,3}

¹ Pacific Northwest National Laboratory, Richland, WA 99352, USA

² Department of Mechanical Engineering, and Institute for Integrated Energy Systems, University of Victoria, Victoria, BC, Canada V8P 5C2

³ Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

Correspondence should be addressed to David P. Chassin; dchassin@uvic.ca

Received 30 January 2014; Revised 12 May 2014; Accepted 12 May 2014; Published 23 June 2014

Academic Editor: Hongjie Jia

Copyright © 2014 David P. Chassin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Simulation of smart grid technologies requires a fundamentally new approach to integrated modeling of power systems, energy markets, building technologies, and the plethora of other resources and assets that are becoming part of modern electricity production, delivery, and consumption systems. As a result, the US Department of Energy's Office of Electricity commissioned the development of a new type of power system simulation tool called GridLAB-D that uses an agent-based approach to simulating smart grids. This paper presents the numerical methods and approach to time-series simulation used by GridLAB-D and reviews applications in power system studies, market design, building control system design, and integration of wind power in a smart grid.

1. Introduction

Recent smart grid technological advances present a new class of complex interdisciplinary modeling and simulation problems that are increasingly difficult to solve using traditional computational methods. Emerging electric power system operating paradigms such as demand response, energy storage, retail markets, electric vehicles, and a new generation of distribution automation systems not only require very advanced power system modeling tools but also require that these tools be integrated with building thermal and control models, battery storage technology models, vehicle charging system models, market simulators, and detailed power system control models. Historically all of these simulation tools were developed independently and each treated the others as a quasi-static boundary condition, an approach that not only limits their effectiveness in evaluating technology impacts over multiple scale and multiple time horizons but which also neglects potentially important coupling effects.

In the case of electricity transmission simulation, PSLF [1], PSS/E [2], and PowerWorld [3] have a longstanding record showing their ability to simulate bulk power systems in

a wide range of conditions. But even with recent improvement to address new conditions such as fault-induced delayed voltage recovery [4], these tools are largely unable to integrate with the wide variety of tools needed to address distribution-level phenomena in a manner that meets the needs for smart-grid technology developers.

Electricity distribution level tools such as SynerGEE [5], WindMil [6], Cymdist [7], and RTDS [8] face similar challenges integrating with wholesale market and renewable integration tools because they too were designed using conventional models that depend on homogeneous descriptions of the underlying electromechanical behavior of the electric power system, either as an electromagnetic transient solution with timescale of microsecond to milliseconds or as a steady-state power flow solution with no timescale at all. At intermediate timescales of seconds, minutes, hours, and days there are many important phenomena that these simulations cannot incorporate.

The same can be said for building energy simulation, battery storage models, market simulations, and distribution automation controls in that they cannot represent the behaviors of the subject systems at the same time and size

scales as power system simulations. Thus, the problem of integrating these tools into a single multiscale computational framework appears insurmountable using conventional simulations based on the numerical solution of systems of ordinary or partial differential equations (or their discretized counterparts) to represent changes in quantities of interest such as the voltage at a bus, the price of energy, the temperature in a building, or the charge in a battery.

Agent-based computational economics [8] has increasingly been applied to electricity markets and consequently was among the first fields to address the challenges of using agent-based tools in power system simulation. The limitations with classical modeling methods [9], the tendency to ignore learning as a result of one-shot auctions [10], and the concerns with stylized trading models used by game theoretic methods [11] are among the chief motivations cited for using agent-based methods. The exploration of multiple equilibria [12] and a change from a focus on rational behavior and equilibrium processes toward heterogeneity and adaptation [13] only becomes possible with significant computing resources. As a result, computational economics has been divided into four areas of investigation [12]:

- (1) empirical research that seeks to understand why and how macroscopic regularity emerges from microscopic properties and behaviors;
- (2) normative research that uses agent-based models as an *in silico* laboratory to design and test policies;
- (3) theory generation that uses structured analysis to discover the conditions under which global regularity evolves;
- (4) methodology development that seeks to improve the tools and methods that support computational economics.

Consistent with the postulate that markets should be designed using engineering tools [14] and anticipating the coming smart grid revolution the US Department Energy's Office of Electricity commissioned Pacific Northwest National Laboratory to develop a simulation environment that would address the gaps in existing power system simulation and modeling tools. The first open-source release of the GridLAB-D [15] occurred in April 2008 and by November 2010 GridLAB-D was used to study a variety of smart grid problems in demand response and renewable integration [16–24]. Since then, the software has been improved and additional capabilities have enabled the study of a wide range of smart grid problems including conservation voltage reduction, microgrid control, retail market design, wholesale-retail market integration, distributed resource control, smart grid technology readiness evaluation, distributed energy resource integration, reduced-order model development, appliance control strategies, generation intermittency impacts on distribution systems, photovoltaic integration impacts, large-scale integration of wind power, smart grid cost-benefit analysis, and transmission-distribution system model integration.

The primary purpose of this paper is to place the development of GridLAB-D in the context of research on the application of agent-based simulations and provide details

of how GridLAB-D solves interdisciplinary simulation problems as a time series using the agent-based paradigm. The first section discusses the general features of agent-based systems and briefly discusses examples and features of such systems in various domains. Next, the solution methods used by GridLAB-D are discussed and application examples are reviewed to demonstrate how the methods have been applied to various smart grid problems. Finally, planned future work is discussed and opportunities for other researchers to contribute further developments to the open source GridLAB-D tools are enumerated.

2. Fundamentals

Agent-based modeling is not a new approach to modeling complex systems. Early development of this approach was pioneered by Von Neumann [25], popularized by Holland [26] and Conway [27], and systematized by Wolfram [28]. But advances in computational capabilities in recent years have made large-scale agent-based models much more accessible to the nonexperts using desktop computing systems. Agent-based simulations have become commonplace in games, finance, epidemiology, ecological research, and training systems, to name a few examples. This section will examine a well-known example from ecology to elucidate the fundamental aspect of agent-based simulation.

To illustrate the difference between conventional models and agent-based models, we review the Lotka-Volterra predator-prey model [29], a well-studied class of system that has been modeled using both conventional and agent-based methods. A Lotka-Volterra system describes a simple ecosystem that exhibits quasi-harmonic behavior we can observe using simulations based on both methods and thus provides a good basis for comparison [30]. This well-known predator-prey system is described by the ordinary differential equations

$$\dot{x} = x(a - by), \quad (1a)$$

$$\dot{y} = y(cx - d), \quad (1b)$$

where x is the size of the prey population at time t and y is the size of the predator population at the same time t . If x represents the number of rabbits and y represents the number of foxes, (1a) says that while rabbits grow at a rate a they are also killed by foxes at a rate proportional to the number of foxes $b \cdot y$. Similarly, (1b) says that while foxes grow as a function of the food supply $c \cdot x$ they also die at a rate d .

The simplicity of the Lotka-Volterra system lends itself to analysis in the sense that one can compute aggregate properties such as the fixed population equilibrium by solving (1a) and (1b) for the steady state when $\dot{x} = \dot{y} = 0$. In this case we find only one nontrivial equilibrium state when

$$y = \frac{a}{b}, \quad x = \frac{d}{c}. \quad (2)$$

Similarly, the stability of the fixed points can be determined using the Jacobian

$$J(x, y) = \begin{bmatrix} a - by & -bx \\ dy & cx - d \end{bmatrix} \quad (3)$$

from which we conclude the trivial fixed point $J(0,0)$ is an unstable saddle point, which explains why populations are not “attracted” to extinction conditions. The nontrivial fixed point is different because $J(d/c, a/b)$ has imaginary eigenvalues $\lambda_1 = i\sqrt{ad}$ and $\lambda_2 = -i\sqrt{ad}$ and no conclusions can be drawn. Solving the original differential equations by integrating directly allows us to find a conserved quantity

$$C = a \ln y(t) - by(t) - cx(t) + d \ln x(t), \quad (4)$$

the value of which corresponds to a stationary population that oscillates around the nontrivial fixed point along invariant trajectories. Thus, satisfying (4) provides the basis for any simulation that will accurately model the population dynamics based on the parameters (a, b, c, d) and the initial conditions $x(0)$ and $y(0)$. Given this condition all possible states of the system can be explored, as shown in Figure 1.

Several significant problems become apparent when one attempts to find analytic solutions to many real-world systems. First, the parametric form of (4) is often difficult to solve numerically as a time-series solution, even for simple systems such as the Lotka-Volterra model, and finite difference methods often exhibit numerical integration errors that accumulate over time and lead to divergence, as shown in Figure 2. The source of this particular error is the estimate of the derivative at the start of each finite time interval. Euler’s method addresses this problem to a first order and higher order solutions using Runge-Kutta methods to eliminate the error. Unfortunately, for many systems these error correction methods can be challenging to implement numerically using suitable finite difference methods.

The second problem is any change to the structure of the system or coupling with other dynamic systems requires that the solution be rederived (often manually) from the original differential equations. This difficulty is encountered when using a mixed electromechanical, thermal, and economic system such as

$$\tilde{I}_{N \times 1} = Y_{N \times N} \tilde{V}_{N \times 1}, \quad (5a)$$

$$\tilde{V}_{L \times 1} \tilde{I}_{L \times 1}^* = Q_{L \times 1} R(P_{M \times 1}), \quad (5b)$$

$$\tilde{V}_{G \times 1} \tilde{I}_{G \times 1}^* = Q_{G \times 1} R(P_{M \times 1}), \quad (5c)$$

$$\frac{\partial D_{L \times M}}{\partial P_{L \times M}} = \frac{\partial S_{G \times M}}{\partial P_{G \times M}}, \quad (5d)$$

where $\tilde{I}_{N \times 1}$ represents the phasor currents flowing into the network at the N electric nodes, $Y_{N \times N}$ is the node admittance matrix, $\tilde{V}_{N \times 1}$ represents the N node voltage phasors, $Q_{L \times 1}$ represents the L customers thermal loads, $Q_{G \times 1}$ represents the G generators outputs, $R(P_{M \times 1})$ represents the generator or customers responses to the M prices for electricity (e.g., energy, power, ramping), $D_{L \times M}$ is the demand of L customers in M markets, and $S_{G \times M}$ is the supply of G generators in M markets.

The first equation describes the equilibrium condition for the electric power flow network, the second and third equations describe the equilibrium condition for the generator and consumer response to pricing, and the fourth equation

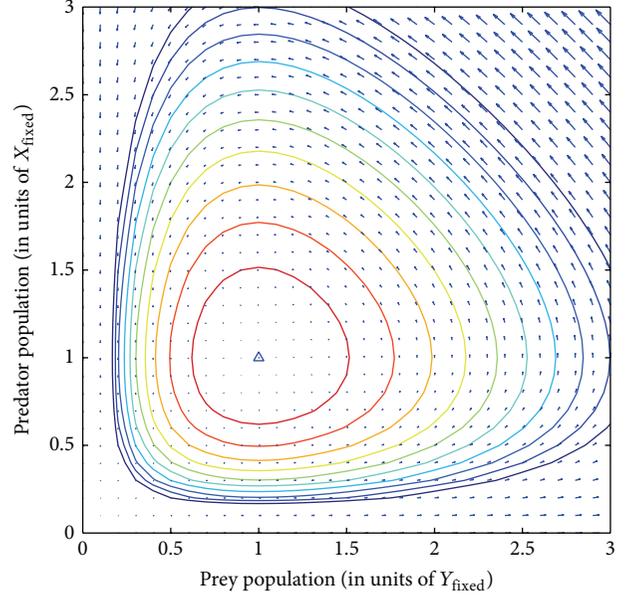


FIGURE 1: State space trajectories of a predator-prey system.

describes the equilibrium condition for the power market. These systems tend to operate on different timescales using different variables to describe interfaces between them.

Finally, the third problem is that as the systems become more complex the differential equations become so numerous and unwieldy that the model becomes analytically intractable for any nontrivial condition. This is certainly the case when power systems, market systems, and building thermal models are combined as above with (5a)–(5d).

Fortunately, no matter how complex these systems become they can be numerically modeled using agent-based methods. To illustrate how this is done we review a model of the same predator-prey systems using Conway’s Game of Life, which uses cellular automata to represent a 2-dimensional landscape in which the populations interact. This landscape is represented by a matrix of cells that can take one of three values: 0 when a cell is vacant; 1 when it is occupied by a prey; and 2 when it is occupied by a predator. To simulate the advance of time the total population of each is counted as the matrix is iteratively updated using simple rules such as the following:

- (1) the probability that a rabbit is born in a cell “adjacent” to a cell occupied by a rabbit is p ;
- (2) the probability that a fox replaces a rabbit in a cell “adjacent” to a fox is q ;
- (3) the probability that a fox dies is r .

Here, the meaning of “adjacent” uses a “north-east-south-west” cell adjacency condition that embodies the probability of the xy component in the Lotka-Volterra model. But the 2-dimensional adjacency definition is unnecessary for the purposes of simulation. Adjacency can also be accomplished using a 1-dimensional “left-right” adjacency definition without loss of generality.

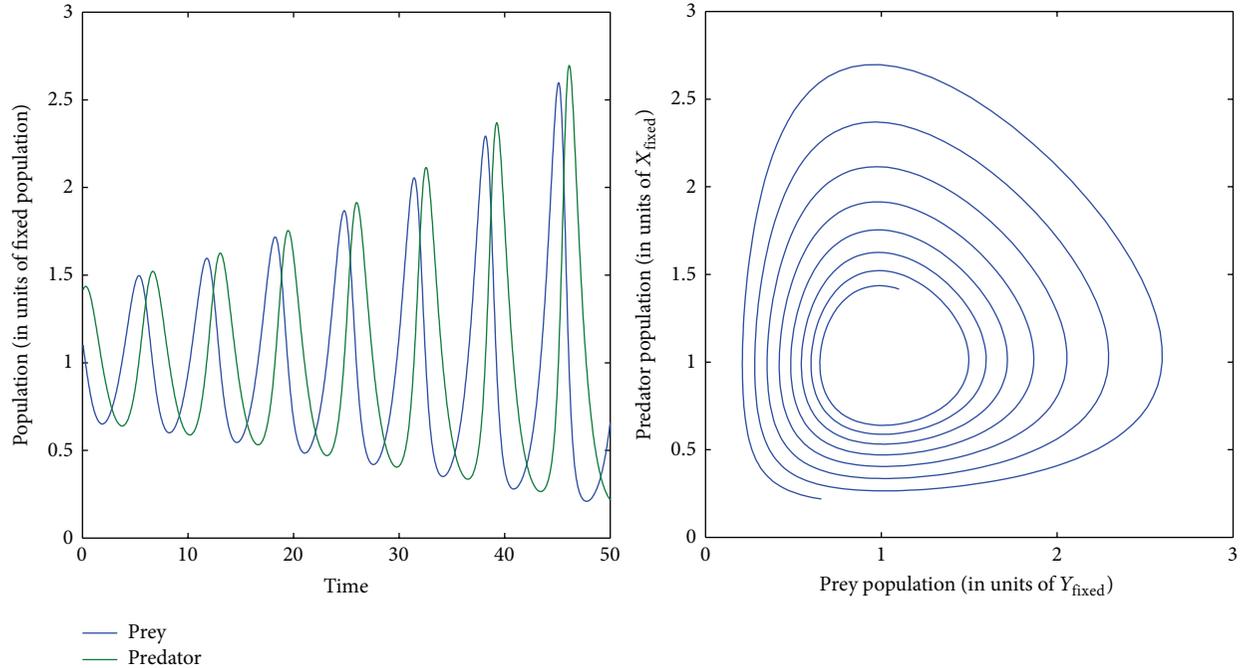


FIGURE 2: Integration error of a “naive” finite difference solution.

The output of an agent-based simulation of the Lotka-Volterra model based on these rules using a simpler random 1-dimensional “encounter” map generated at each iteration can be seen in Figure 3 for $p = 0.2$, $q = 1.0$, and $r = 0.2$. The fixed point for these conditions is $x = 10^4$ and $y = 5 \times 10^3$. The simulation produces similar oscillatory behavior to that observed in the analytic model. The model naturally introduces small fluctuations to which it is sensitive, which is why it deviates from the fixed point even when it is initialized at it. This phenomenon can be important in systems where the action of a single entity can influence the outcome of the entire system. Such situations are difficult to describe using ODEs because they involve fast-growing instabilities emanating from independent perturbations.

The simulation shown in Figure 3 exhibits another characteristic not seen in ODE solutions and which is shown in Figure 4 (only the initial conditions have changed). While finite difference solutions typically exhibit distinctly convergent or divergent behavior, agent-based solutions often exhibit mixed convergence behavior not seen in simpler ODE solutions.

The primary characteristics of verisimilar agent-based models are based on realistically defining the agents and their relationships in an interaction landscape so that their evolution over time accurately reproduces the dynamics of the system being modeled. This requires that the following considerations be addressed carefully:

- (1) the internal states of the agents are represented by variables (discrete or continuous) that provide sufficient dynamic range and resolution to allow small fluctuations to affect agent behavior realistically;

- (2) the agents’ behaviors are represented such that they evolve in a manner akin to a state machine (e.g., a Markov process, a cellular automaton, or a state space model) or equivalent model (e.g., differential equations);

- (3) the agents interact with an environment that evolves over time such that the agents are both affecting the environment and affected by the environment;

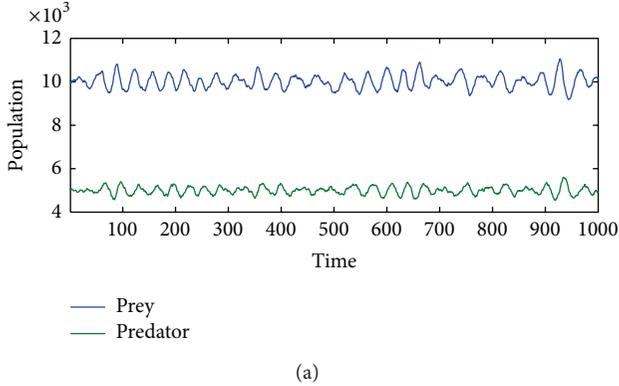
- (4) the agents interact with each other in a manner that is consistent with the expected interactions in the system being modeled; that is, not all the internal state variables of agent are revealed to other agents.

More complex systems may also involve the following additional considerations:

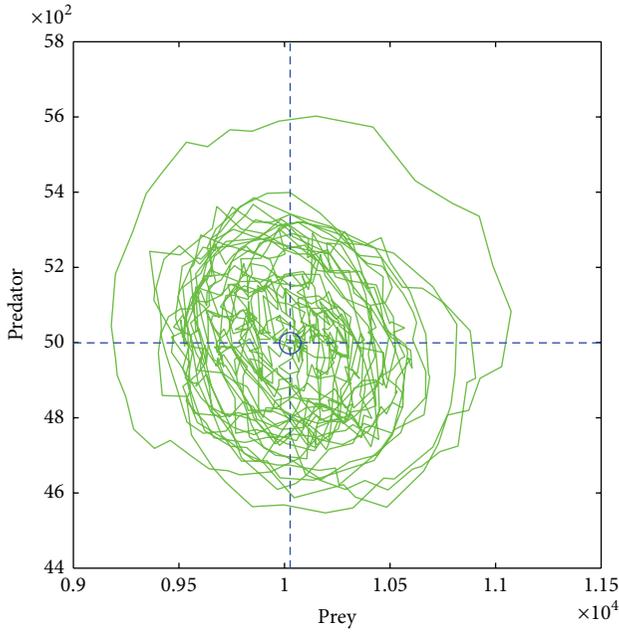
- (5) the environment may change over time; that is, an external simulation, underlying model, or pre-recorded set of conditions changes it slowly with respect to the dynamics of the agents;

- (6) agents of different types can be interacting concurrently, which is the most common situation in highly realistic simulations.

Unfortunately, while the descriptive power of agent-based models is readily apparent and has been amply demonstrated [31–34], these models have a few significant shortcomings that remain for the most part unresolved. The first is that agent-based simulations provide even less analytic insight than numerical simulations that are derived on *ab initio* models. For example, the simulation shown in Figures 3 and 4 has an obvious fixed point that corresponds to what we expect from the Lotka-Volterra model. However, the simulation does

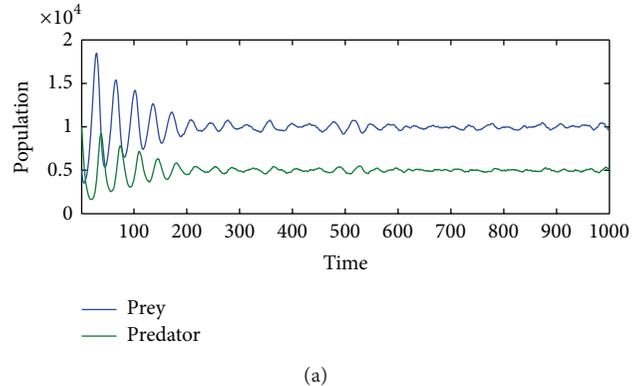


(a)

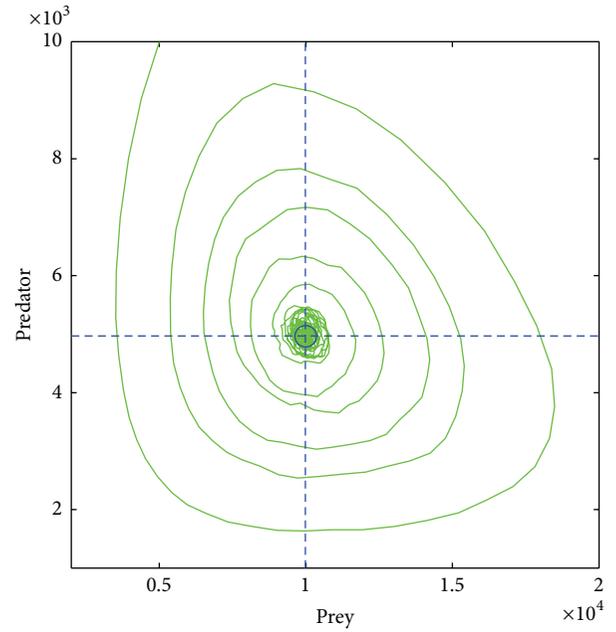


(b)

FIGURE 3: Fluctuation behavior of agent-based simulation.



(a)



(b)

FIGURE 4: Convergent behavior of agent-based simulation.

not provide us with an analytic relationship between the fixed point that we observe, which is easily found numerically by taking the mean values of x and y over a nontrivial range of time, and the parameters of the analytic model or the probabilities of the rules. While both x and y can be determined analytically from the Lotka-Volterra parameters using (2), there is no obvious way to relate the fixed point to probabilities p , q , and r without running the agent-based model. This challenge remains unresolved except for the most trivial system.

The second shortcoming is that agent-based model verification and validation are difficult to accomplish using formal methods. When considering conventional simulations such as the ODE solution to the Lotka-Volterra system, verification is the process of ensuring, for example, that the populations always satisfy (4) given any particular initial conditions $x(0)$ and $y(0)$, whereas validation is the process of ensuring that a series of observations of a real population evolves in a manner consistent with predictions of the simulation.

In principle the same should be possible with agent-based simulations. However, as we have already seen agent-based models often exhibit fluctuations that resemble real systems so that instead of trying to verify an idealized model (nonfluctuating) against real (fluctuating) data, model developers are often trying to validate the model using a system that fluctuates in a different way. The verification question is no longer just about the uncertainty associated with the empirical data. Now the uncertainty associated with the agent-based model must be considered as well. Even for simple models like the Lotka-Volterra system the agent-based model does not strictly obey its conservation law: it only approximately follows the law in the sense that the simulation converges toward the fixed point when far from it but diverges when very close to it.

The problems with agent-based model validation came to the fore in the development of agent-based computational economics used in the design of electricity markets. In their discussion of this problem LeBaron and Tesfatsion identified

three challenges [37]. Foremost is the embarrassing number of degrees of freedom that arise from the large number of parameters contained in the models (a simple 1000-home GridLAB-D model contains nearly 200,000 distinct parameters that can affect the outcome). This problem is aggravated by the nearly unlimited functional and learning algorithms that can be implemented, and even mixed. Finally the properties of the agents themselves are often difficult to identify precisely and are often informed more by modelers intuition and engineering expertise than by data collected or observed human behavior.

One commonly used approach is verification against experimental and/or reduced simulations and validation against empirically collected data that demonstrate consistency with known initial conditions and outcomes. In this way, GridLAB-D models are often verified with simple “known good” simulations and validated using telemetry from large-scale real-world systems for which models are available. The latter is discussed further in the applications section below.

These problems are not unique to GridLAB-D. Widely used agent-based simulation environments such as SWARM [31], Repast [32], EMCAS [33], AMES [34], and others have all addressed these challenges and the reader is referred to these for details on how the verification and validation problem is addressed variously by them.

Ultimately the decision whether to accept the mathematics of agent-based models hinges on an argument made by LeBaron and Tesfatsion in their assessment of the relevant differences between classical and constructive mathematics [37]. The former is supported by those who accept the law of excluded middle so that existence proofs by contradiction are permissible, while the latter is supported by those who require direct proof that a proposition is true in order to rule out both falseness *and* undecidability. This latter proof can be realized as computer programs that embody concepts of information flow that limit what an agent knows and when it knows it:

“This distinction provides a dramatically different perspective on how we perceive models in our mind in relation to the real-world systems they are intended to represent. For example, social system modelers using classical mathematics typically assume (explicitly or implicitly) that all modeled decision makers share common knowledge about an objective reality, even if there is no constructive way in which these decision makers could attain this common knowledge. In contrast, social system modelers advocating a constructive mathematics approach have argued that the “reality” of each modeled decision maker ought to be limited to whatever that decision maker is able to compute.”

GridLAB-D was designed and implemented with the latter view in mind.

3. Solution Method

The success of GridLAB-D as a tool to study smart grids is primarily attributed to the use of the agent-based simulation

paradigm. The approach has made GridLAB-D easy to use in spite of the extensive use of multidisciplinary elements in various modules. In addition the output of GridLAB-D is highly similar to data collected from smart grid demonstration project conducted in the field, which has facilitated verification and validation.

GridLAB-D allows modelers to choose which of the agent-based characteristics are implemented in a given module. Multiple modules may be operated concurrently in any given simulation and there is no requirement that every module uses the same modeling method in any given simulation. For example, the power flow module uses a state-space model with an underlying algebraic solver to compute the voltages and currents given the loading conditions presented to it by the other modules. The residential building module in turn uses the voltages to determine the energy input to home energy systems and an underlying ODE thermal model to solve the indoor air and mass temperatures. The market models use the building control systems to determine the price of electricity, which in turn is used to determine the price at which overall supply and demand for electricity are equal.

Agents are organized into ranks based on the relationships between them imposed on them by the modules’ solvers. The ranks are organized in trees of parent-child relationships, with each parent agent primarily depending on the values accumulated from one or more child agents. The ranks are given ordinal numbers with the greatest ordinal assigned to the agent that is the topmost rank and zero assigned to the bottom-most rank. In practice it is typical to find only one agent at the topmost rank and a plurality of agents of rank zero.

The determination of the ranks is made by the modules and based on the solution method implemented. For example, the *power flow* module uses a different rank structure depending on whether the forward-back sweep method [38] and current injection [39] method are used. When the forward-backsweep method is used, the rank structure tends to require many ranks, whereas when the current injection method is used only two ranks are required. This difference is known to influence the relative performance of the solvers depending on the size and general structure of the electric network being modeled.

To illustrate how GridLAB-D’s solver gathers values from multiple agents consider how the average

$$y = \frac{1}{N} \sum_{n=1}^N x_n \quad (6)$$

is computed. There are three steps required to complete this operation:

- (1) set $y = 0$ and $N = 0$;
- (2) add each x_n to y for $n = 1, 2, \dots, N$ incrementing N each time;
- (3) divide y by N if N is nonzero.

This process can be conducted for N values of x in three phases, within which multiple operations (if any) can be

conducted in parallel provided that the operations on y and N are atomic.

Phase 1

$$\begin{aligned} y &\leftarrow 0 \\ n &\leftarrow 0. \end{aligned} \quad (7)$$

Phase 2

$$\begin{aligned} y &\leftarrow y + x_1, & n &\leftarrow n + 1 \\ y &\leftarrow y + x_2, & n &\leftarrow n + 1 \\ & & & \vdots \\ y &\leftarrow y + x_N, & n &\leftarrow n + 1. \end{aligned} \quad (8)$$

Phase 3

$$y \leftarrow \begin{cases} \text{indet:} & n = 0 \\ \frac{y}{n} & n > 0. \end{cases} \quad (9)$$

In GridLAB-D parlance this process is conducted in the following three parallelized phases. (1) *Pre-Top-Down Pass*: This phase gives agents the opportunity to prepare to receive updates from other agents. (2) *Bottom-Up Pass*: This phase gives agents the opportunity to update other agents. (3) *Post-Top-Down Pass*: This phase gives agents the opportunity to compute final values based on updates received from other agents.

In addition, GridLAB-D includes modules that allow implementation of the so-called “precommit” and “commit” passes before and after the three main phases, respectively. This permits solvers that collect global data to prepare and commit the global data that are affected, if any. There is also a finalize pass that is complete only when the clock is advanced to allow any objects that need to compute time-dependent updates to do so before the clock is advanced.

A number of built-in properties with special characteristics are provided to represent physical or stochastic processes and maintain endogenous or exogenous relationships. These are always updated before the precommit stage to ensure that all the global, module, and object properties are correct at a given time indicated by the global clock. These built-in properties are updated in the following order:

- (i) links to external simulations (both read and write);
- (ii) random variables (based on the supported distributions provided by GridLAB-D);
- (iii) scheduled values (updates on the ISO “cron” schedule standard);
- (iv) load shapes (primarily used to shape values in time using queues, pulse-width modulation, amplitude modulation, or simple analog shapes);
- (v) transforms (functions that update a property based on the values of other properties);

(vi) end uses (structures that describe how electric loads are composed);

(vii) heartbeats (events that occur on a regular basis independent of synchronization events).

All these updates (with the exception of finalize updates) return a time for the next expected event for the object or property in question.

3.1. Convergence and Synchronization. The current implementation of GridLAB-D does not guarantee convergence in the overall solution. Modelers can create situations in which two or more agents cannot find a combination of states that satisfy their respective convergence criteria. The design of GridLAB-D’s overall solver assumes that such situations are not intentional but are due to modeling error. In this case GridLAB-D detects the resulting large iteration count without the global clock changing and stops the simulation. Modelers must implement noniterative methods of resolving such state conflicts based on the assumption that they occur at a time scale less than the time resolution of the main solver.

If necessary modelers can process events with a time resolution less than 1 second using a discrete fixed time-step solution method. The discrete time step used is the shortest time step requested by the agent(s) seeking subsecond processing. During processing of subsecond simulations event-driven simulation is disabled until all agents indicate that they no longer require subsecond processing, which typically occurs when the transient behavior settles to steady state and event-based processing can resume.

Parallelization of many event and time-step update computations is accomplished using a “thread group” strategy. This multiprocessor approach improves simulation performance by preallocating computational threads to groups of objects that can always be executed independently. The determination of parallelizability is based on the rank of an object and the type of event being processed. Ranks are established on the basis of which way information flows during synchronization events, if any, with high-rank objects depending on multiple low-rank objects during the bottom-up synchronization event. This method of parallelization has been shown to exhibit approximately linear scaling for the smaller number of computing units typically found in desktop computing systems [40].

Synchronization avoidance strategies are also included in GridLAB-D’s main solver to reduce the number of unnecessary events and improve overall simulation performance. Among these is the “valid to” time, which agents may set when the time of the next event in an agent is independent of external inputs to the agent and the agent is not expecting to make any changes to its internal state until that time. Using this valid-to time the main solver can avoid processing certain agent event when the outcome of the call is a forgone conclusion and the agent is not expected to change state.

3.2. Standard Modules and Solution Methods. The AC power flow solution is implemented using a modified Newton-Raphson method for meshed electric networks [41]. The

power flow solver supports unbalanced three-phase networks. The modeler does not need to directly compute the admittance matrix as the solver performs this update during synchronization from the properties of the objects that represent the various electrical components supported by GridLAB-D power flow module, including power lines, transformers, switchers, capacitor banks, and voltage regulators. In cases where the electric system is radial the modeler may opt to use the power flow module's forward-back sweep solver, which is based on Kersting's method [42]. A separate generators module provides classes that allow modelers to implement various electricity generating and storage resources.

Building thermal response is solved using the equivalent thermal parameters (ETP) method, which implements both the time and temperature solutions to the second-order ordinary differential equation that describes the response of the indoor air temperature of a building to outdoor temperature conditions, internal appliance and occupant heat gains, ventilation gains/losses, solar gains, and heating/cooling system state. The building modules include thermostatic controllers and appliance models, some of which incorporate demand response control strategies. A retail electricity market is included that implements a double-auction for feeder capacity and determines the real-time price (RTP) at which feeder supply is equal to the total load. The retail market supports both demand response resources, such as thermostats and electric vehicle chargers, as well as distributed generation resources such as diesel backup generators, microturbines, photovoltaics, and energy storage devices.

4. Applications

GridLAB-D has been used to study a wide variety of power system problem as summarized above. In this section we examine a few of the results obtained in more detail and discuss the role that GridLAB-D's solution method played in enhancing the analysis beyond what is possible using conventional simulation tools. Specific applications or recent interests include Volt-VAR optimization (VVO), dynamic real-time pricing (RTP) experiments, and integration of renewable energy aided by demand response. In some cases, reliable solutions can be found from other methods, while in others, the agent-based methodology provides unique insight. As validation is often a key question in agent-based systems, studies that have been validated against experimental field data will be discussed.

4.1. Volt-VAR Optimization. VVO is a traditional utility technique for reducing energy consumption or peak demand on an electric circuit by lowering the system voltage to the lower portion of the operational voltage band [43]. With the proliferation of communication and sensor equipment, the control and optimization techniques have become increasingly more complex, but there are always questions about the tradeoff between the cost of a more complicated system and the benefits. GridLAB-D was used to estimate these

benefits prior to deployment [44], and simulations crossed the boundary between power system and load behavior.

Each of the loads within the system is modeled as a complex process driven by inputs such as outside air temperature, occupancy, and thermostat set points. Each load is modeled as an agent with its own specifications and inputs. As the voltage varies throughout the distribution system, this is also used as an input into the load model to quantify how a change in system voltage affects the energy consumption of the individual devices, and thus the overall system. For example, an electric water heater is essentially a resistive element at any given moment in time. As voltage is reduced, the power demand decreases. However, because of the closed-loop thermostatic control on the device, the same amount of energy is required to heat the water. This affects the duty-cycle behavior of the water heater and is tracked through time by each agent. The effect is that peak load (i.e., the maximum number of devices in the *on* state simultaneously) is reduced, but energy consumption (i.e., the cumulative time the loads are in the *on* state) is not. Each model, whether an air conditioner, dishwasher, or other appliance, has its own inputs and responses to changes in voltage. Traditional models, which are basically linear, time-invariant solvers, are not able to easily capture these effects.

In a study with American Electric Power (AEP), new VVO technology was tested in GridLAB-D on eight distribution circuits. The technology was then deployed and tested on those same circuits. Simulation predicted a 2.9% average reduction while deployment produced a 3.3% average reduction in energy consumption. However, there were significant differences on individual feeders that could be attributed to changes in load composition between the modeling and testing phases.

4.2. Real-Time Pricing Demonstration. As part of the American Recovery and Reinvestment Act, a real-time pricing experiment was devised to engage consumer loads with five-minute energy prices, in-home displays, and equipment utilizing automated response [45]. The automated system requires individual air conditioners to construct a market bid that reflects their desire to run during the next market period (every five minutes). A central auction collects all of the bids, clears a market price, and then broadcasts a (single) price signal to all of the devices. In turn, the devices respond to this signal in a coordinated (but noncommunicative) manner by modifying the behavior of the thermostat. The end goal is to reduce overall energy costs, both for the customer and the utility, and to reduce demand when there is a physical constraint on the system. GridLAB-D was used to develop and fine-tune the control and communication requirements prior to deploying this system [46].

The power system is modeled along with individual appliance loads. In addition, each thermostat is modeled with additional market controls acting as bidding agents, and a centralized auction agent collects all of the information and dispatches a price signal. When required, the communication system is also modeled separately (i.e., message packets are dispatched through a communication layer rather than

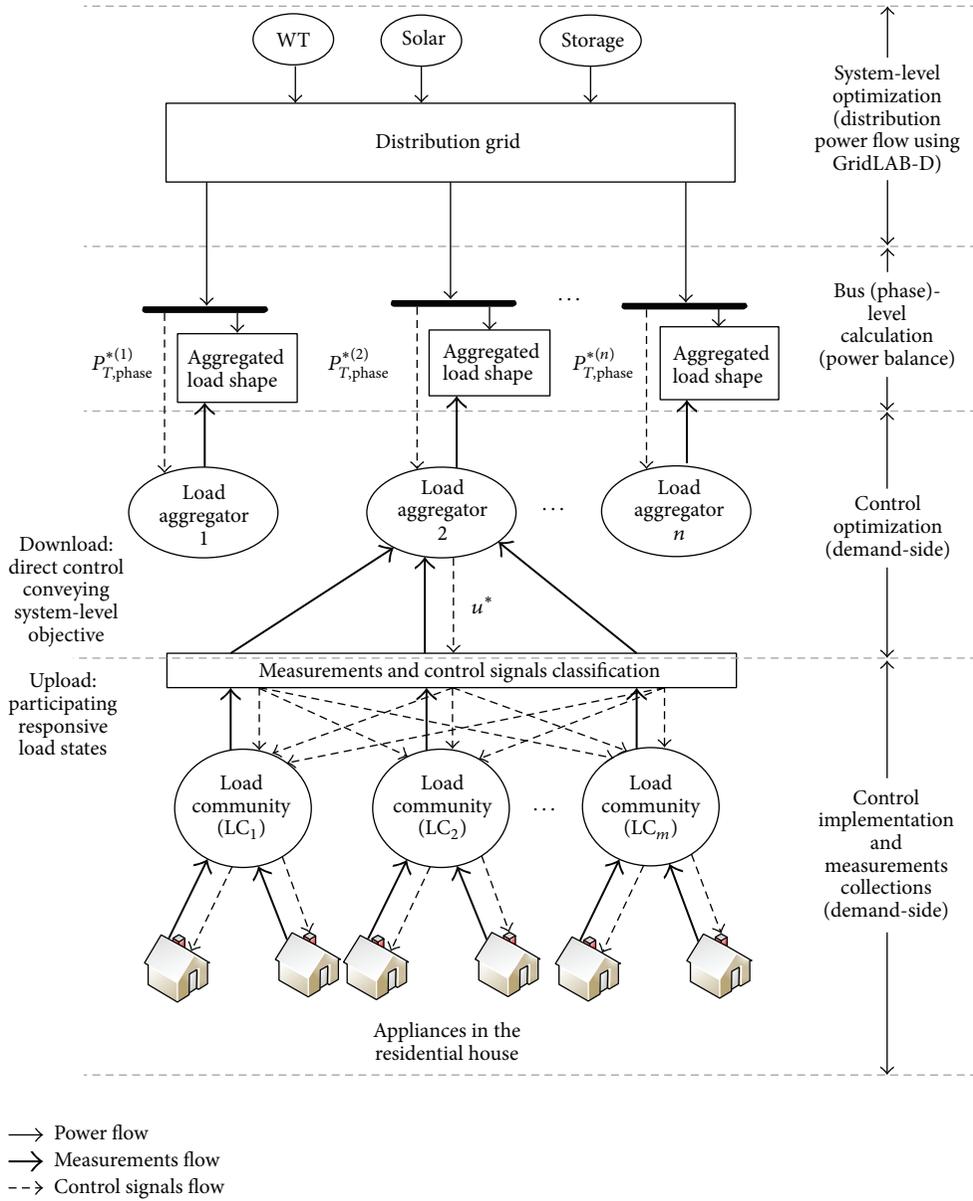


FIGURE 5: GRIDLAB-MATLAB modelling framework for a smart self-regulating system (reproduced from [35] with permission from Elsevier).

directly within the software), including message delays and dropped packets [47]; each of the agents is responsible for understanding what to do when information is lost or delayed.

By working through the design of the control system (both the distributed bidding agents and the centralized auction) via the simulator prior to deployment, a number of effects and impacts could be evaluated and redesigned. Effects that caused undue strain on the system (such as synchronization of loads in response to the price signal, errors in load prediction, or loss of data) and potential impacts (such as changes in customer bills, equitable rebate and incentive mechanisms, or violation of local constraints) were evaluated and modified. For example, it was found through simulation that a slight error in the agent bidding

caused by the thermostat deadband could cause system oscillations in power demand under certain circumstances. This was potentially a major flaw in the control system and was corrected prior to construction of the thermostats that were deployed. Reports comparing field demonstration results to GridLAB-D simulations will be available in late 2014. The combination of linear and nonlinear solutions, noncontinuous state variables, binary operations, and sorting algorithms is not solvable through direct solution methodologies.

4.3. Demand Response for Renewable Integration. Environmental concerns have spurred a significant growth of electricity generation from wind power and other renewable energy

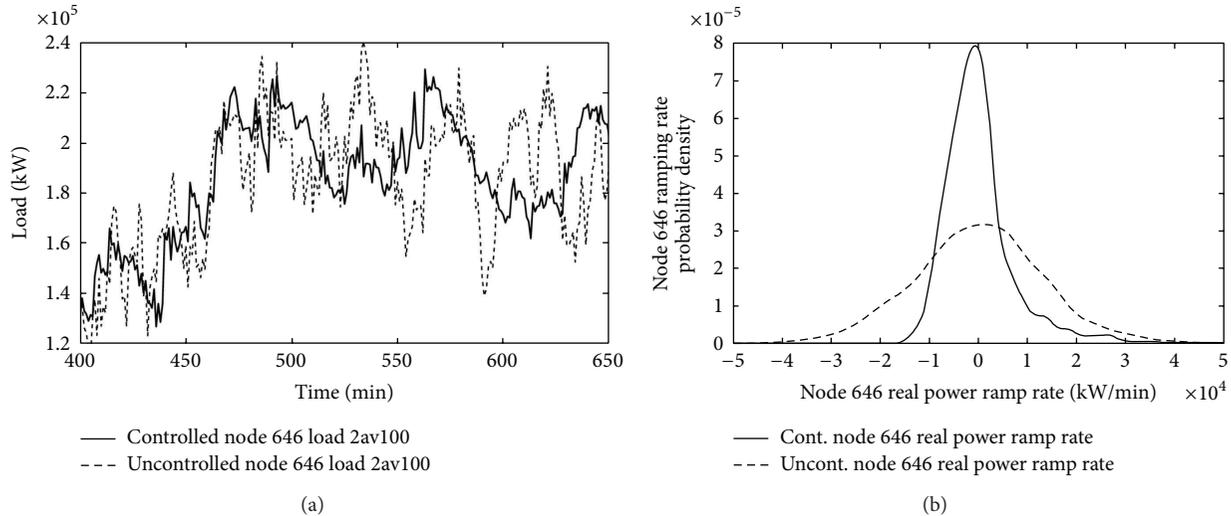


FIGURE 6: Effect of control on (a) load and (b) real power ramp rate (reproduced from [36] with permission from Elsevier).

sources in the last decade. The temporal and spatial variability of these resources present a number of challenges to power system operators, particularly with respect to power system reliability and reserve requirements. These technoeconomic challenges have typically limited wind penetration to at most 30% of grid generation. Smart grid technology now makes it possible to address these challenges using novel strategies such as demand response, whereby loads can be controlled in response to power imbalance or market price signals, and adjust their power demand. This essentially shifts part of the burden of balancing power from the supply to the demand side and results in a reduction of costly contingency reserves.

Using GridLAB-D in conjunction with MATLAB, Williams presented a smart grid model to assess the potential of mitigating fluctuations associated with distributed wind power by using self-regulating, thermostatically controlled heat pumps. The modeling framework for the smart self-regulating system is shown in Figure 5.

Different bus-level control algorithms were investigated using the model, and Figure 6(a) illustrates the effectiveness of bus-level distributed heat pump management strategy in reducing load flow fluctuations by adjusting the heat pump demand to follow wind generation. A useful way of measuring the level of mitigation of wind fluctuations is to examine the required ramping rates spectrum of probability distribution [48]. Figure 6(b) shows the significantly lower ramp rates achieved by the control strategy.

The viability of demand response will require effective market pricing and the integration of price signals and load controls. Broer et al. [35] developed a general simulation framework integrating a GridLAB-D smart grid system with a market model. The model incorporates generator and load controllers and allows bidding from both the supply and demand sides into a double-auction RTP electricity market. Demand response in the system is achieved through

thermostatically controlled loads such as heating, ventilation, and air-conditioning (HVAC) units and electric water heaters.

This model was validated using a physical demonstration project conducted in the Olympic Peninsula, Washington, USA. RTP simulation results obtained using a system comprising 10,000 residential houses and a grid-integrated 35 MW wind park are shown in Figure 7. The wind and hydrosupplies consistently bid at $\$0/\text{kWh}$ and $\$0.1/\text{kWh}$, respectively. The figure shows the response of a sample residential house to wind power variations. Bidding on the demand side is from the responsive HVAC load. The HVAC load switches off when the bid is below the clearing price. This happens when the clearing price rises as a result of decreasing wind power. The HVAC system switches on again when wind power recovers. This type of model allows operators to assess the impact of extreme scenarios, such as the persistence of high/low wind regimes over extended periods, and during which a diversity of loads needs to be maintained in order to avoid saturation (all loads becoming unresponsive).

These GridLAB-D-based renewable energy integration studies indicate that controlled customer power consumption can be modified to facilitate wind energy integration without compromising customers comfort. Such DR strategies can effectively modify load flow, improve energy efficiency, and reduce contingency reserve requirements. The versatile GridLAB-D agent-based modelling provides an integrated framework to assess the potential of various demand response strategies and to support the design of virtual power plants that can effectively provide the additional contingency reserve and regulation capacity required to increase the penetration of variable renewable energy generation in the electricity grid. Again, this cannot be accomplished through the use of traditional solution methods without making gross over-simplifications of the underlying behavior and response systems.

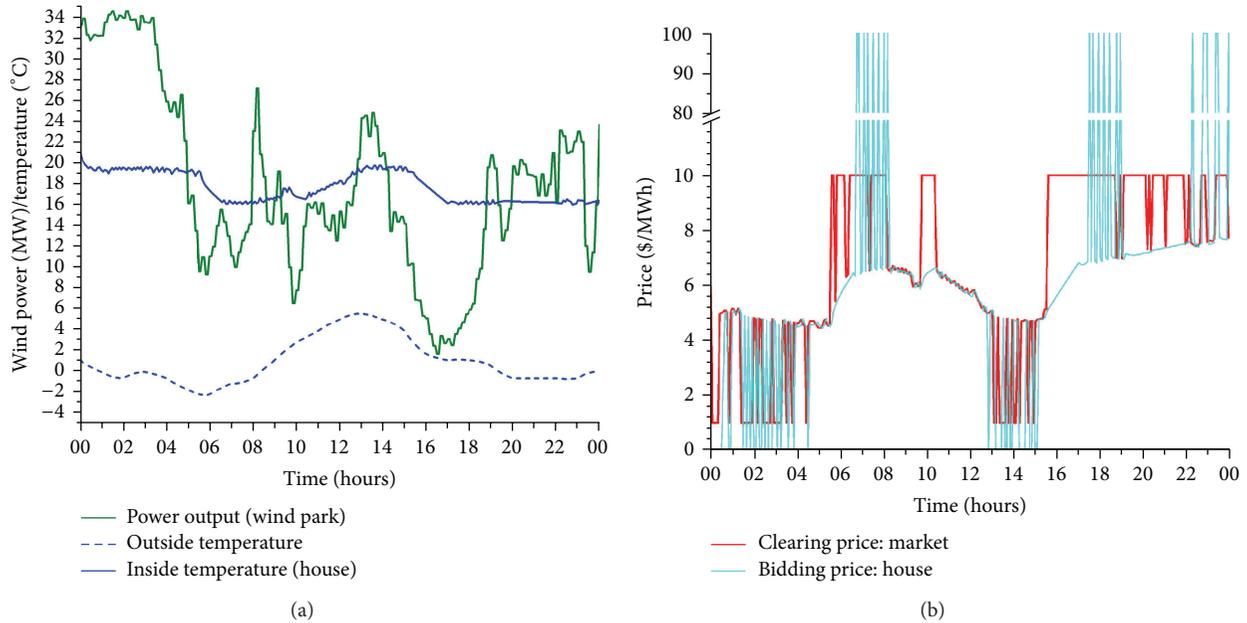


FIGURE 7: Controlled behaviour of an individual house over a 24-hour period in response to varying wind power. (a) Indoor house temperature following wind power. (b) Variation of market clearing price and resulting turning off of loads as a result of changes in wind power (reproduced from Broerer et al. [35] with permission from Elsevier).

5. Future Work

As an open-source tool for the smart-grid research community there are many prospective contributors to GridLAB-D development and thus many directions where it can go. The US Department of Energy's Office of Electricity, which currently directs the development of GridLAB-D at Pacific Northwest National Laboratory (PNNL) is committed to ongoing improvements in the power system, building, markets, controls, and telecommunications modules themselves. In addition, PNNL is making investments in internal solution methods, with special consideration being given to parallelization for high-performance computing platforms, cosimulation environments to simplify integration with multiple simulation environments, and improvements to allow more formal model verification and validation methods.

6. Conclusions

In this paper we have presented previously unpublished details on the agent-based simulation methods implemented in GridLAB-D. Our objective is in part to introduce the applied mathematics community to the challenges faced by those who employ agent-based methods and encourage greater collaboration between applied mathematics and power engineering communities. We have presented the rationale for adopting an agent-based simulation approach to simulating the smart grid, discussed some of the challenges with using agent-based methods to design interdisciplinary smart-grid technology solutions, and reviewed at a high-level some applications and studies that best exemplify the versatility and impact of GridLAB-D.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was funded in part by the US Department of Energy's Office of Electricity.

References

- [1] "General Electric PSLF Software," <http://www.gepower.com>.
- [2] PTI Siemens, 'PSS' E 30.2 Program Operational Manual, 2011.
- [3] "PowerWorld Simulation software," <http://www.powerworld.com>.
- [4] D. Kosterev and D. Davies, "System model validation studies in WECC," in *Proceedings of the IEEE PES General Meeting (PES '10)*, July 2010.
- [5] "SynerGEE Electric software," <http://www.gl-group.com>.
- [6] "WindMil software," <http://www.milsoft.com>.
- [7] "Cymdist software," <http://www.cyme.com>.
- [8] R. Kuffel, J. Giesbrecht, T. Maguire, R. P. Wierckx, and P. McLaren, "RTDS: a fully digital power system simulator operating in real time," in *Proceedings of the IEEE WESCANEX Communications, Power, and Computing Conference (WESCANEX '95)*, vol. 2, pp. 300–305, May 1995.
- [9] M. Ventosa, Á. Baíllo, A. Ramos, and M. Rivier, "Electricity market modeling trends," *Energy Policy*, vol. 33, no. 7, pp. 897–913, 2005.
- [10] M. H. Rothkopf, "Daily repetition: a neglected factor in the analysis of electricity auctions," *Electricity Journal*, vol. 12, no. 3, pp. 60–70, 1999.

- [11] R. Wilson, "Architecture of power markets," *Econometrica*, vol. 70, no. 4, pp. 1299–1340, 2002.
- [12] L. Tesfatsion, "Agent-based computational economics: a constructive approach to economic theory," in *Handbook of Computational Economics*, vol. 2, pp. 831–880, 2006.
- [13] R. M. Axelrod, *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*, Princeton University Press, 1997.
- [14] A. E. Roth, "The economist as engineer: game theory, experimentation, and computation as tools for design economics," *Econometrica*, vol. 70, no. 4, pp. 1341–1378, 2002.
- [15] "GridLAB-D software," <http://www.gridlabd.org/>.
- [16] R. T. Guttromson, D. P. Chassin, and S. E. Widergren, "Residential energy resource models for distribution feeder simulation," in *Proceedings of the IEEE Power Engineering Society General Meeting*, vol. 1, pp. 108–113, July 2003.
- [17] S. E. Widergren, J. M. Roop, R. T. Guttromson, and Z. Huang, "Simulating the dynamic coupling of market and physical system operations," in *Proceedings of the IEEE Power Engineering Society General Meeting*, vol. 1, pp. 748–753, June 2004.
- [18] N. Lu and D. P. Chassin, "A state-queueing model of thermostatically controlled appliances," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1666–1673, 2004.
- [19] N. Lu, D. P. Chassin, and S. E. Widergren, "Modeling uncertainties in aggregated thermostatically controlled loads using a state queueing model," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 725–733, 2005.
- [20] J. M. Roop, E. M. Fathelrahman, and S. E. Widergren, "Price response can make the grid robust: an agent-based discussion," in *Proceedings of the IEEE Power Engineering Society General Meeting*, vol. 3, pp. 2813–2817, June 2005.
- [21] D. P. Chassin, K. Schneider, and C. Gerkenmeyer, "GridLAB-D: an open-source power systems modeling and simulation environment," in *Proceedings of the Transmission and Distribution Exposition Conference*, April 2008.
- [22] D. P. Chassin, K. Schneider, and C. Gerkenmeyer, "GridLAB-D: an open-source power systems modeling and simulation environment," in *Proceedings of the Transmission and Distribution Exposition Conference*, April 2008.
- [23] K. P. Schneider, D. Chassin, Y. Chen, and J. C. Fuller, "Distribution power flow for smart grid technologies," in *Proceedings of the IEEE/PES Power Systems Conference and Exposition (PSCE '09)*, Seattle, Wash, USA, March 2009.
- [24] D. P. Chassin and S. E. Widergren, "Simulating demand participation in market operations," in *Proceedings of the IEEE Power and Energy Society General Meeting (PES '09)*, July 2009.
- [25] J. Von Neumann, *Theory of Self-Reproducing Automata*, 1966.
- [26] J. H. Holland, "Studies of the spontaneous emergence of self-replicating systems using cellular automata and formal grammars," in *Automata, Languages, Development*, pp. 385–404, North-Holland, Amsterdam, The Netherlands, 1976.
- [27] J. Conway, "The game of life," *Scientific American*, vol. 223, no. 4, article 4, 1970.
- [28] S. Wolfram, *A New Kind of Science*, vol. 5, Wolfram Media, Champaign, Ill, USA, 2002.
- [29] A. J. Lotka, *Elements of Physical Biology*, Williams & Wilkins, Baltimore, Md, USA, 1925.
- [30] V. Volterra, *Variazioni e Fluttuazioni del Numero d'Individui in Specie Animali Conviventi*, 1926.
- [31] F. Luna and B. Stefansson, Eds., *Economic Simulations in Swarm: Agent-Based Modelling and Object Oriented Programming*, vol. 14, Springer, 2000.
- [32] N. Collier, *Repast: An Extensible Framework for Agent Simulation*, vol. 36, The University of Chicago's Social Science Research, 2003.
- [33] M. North, C. Macal, G. Conzelmann, V. Koritarov, P. Thimmapuram, and T. Veselka, "Multi-agent electricity market modeling with EMCAS," Argonne National Laboratory, 2004, <http://www.ipd.anl.gov/anlpubs/2002/09/44048.pdf>.
- [34] J. Sun and L. Tesfatsion, "Dynamic testing of wholesale power market designs: an open-source agent-based framework," *Computational Economics*, vol. 30, no. 3, pp. 291–327, 2007.
- [35] T. Broeer, J. Fuller, F. Tuffner, D. Chassin, and N. Djilali, "Modeling framework and validation of a smart grid and demand response system for wind power integration," *Applied Energy*, vol. 113, pp. 199–207, 2014.
- [36] T. Williams, D. Wang, C. Crawford, and N. Djilali, "Integrating renewable energy using a smart distribution system: potential of self-regulating demand response," *Renewable Energy*, vol. 52, pp. 46–56, 2013.
- [37] B. LeBaron and L. Tesfatsion, "Modeling macroeconomies as open-ended dynamic systems of interacting agents," *The American Economic Review*, vol. 98, no. 2, pp. 246–250, 2008.
- [38] W. H. Kersting, *Distribution System Modeling and Analysis*, CRC press, 2012.
- [39] P. A. N. Garcia, J. L. R. Pereira, S. Carneiro, and V. M. Da Costa, "Three-phase power flow calculations using the current injection method," *IEEE Transactions on Power Systems*, vol. 15, no. 2, pp. 508–514, 2000.
- [40] S. Jin and D. P. Chassin, "Thread group multithreading: accelerating the computation of an agent-based power system modeling and simulation Tool-GridLAB-D," in *Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS '14)*, January 2014.
- [41] P. Kundur, *Power System Stability and Control*, McGraw-Hill, 1994.
- [42] W. Kersting, *Distribution System Modeling and Analysis*, CRC Press, 2002.
- [43] "American National Standard for Electric Power Systems and Equipment-Voltage Ratings (60 Hertz)," ANSI C84. 1-2011, 2011.
- [44] K. P. Schneider and T. F. Weaver, "Volt-VAR optimization on American Electric Power feeders in Northeast Columbus," in *Proceedings of the IEEE PES Transmission and Distribution Conference and Exposition (T&D)*, May 2012.
- [45] S. E. Widergren, K. Subbarao, J. C. Fuller et al., "AEP Ohio gridSMART Demonstration Project Real-time Pricing Demonstration Analysis," PNNL Report 23192, Pacific Northwest National Laboratory, Richland, Wash, USA, 2014.
- [46] K. P. Schneider, J. C. Fuller, and D. Chassin, "Analysis of distribution level residential demand response," in *Proceedings of the IEEE/PES Power Systems Conference and Exposition (PSCE '11)*, pp. 1–6, March 2011.
- [47] J. C. Fuller, S. Ciraci, J. A. Daily, A. R. Fisher, and M. Hauer, "Communication simulations for power system applications," in *Proceedings of the Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES '13)*, pp. 1–16, May 2013.
- [48] J. Apt, "The spectrum of power from wind turbines," *Journal of Power Sources*, vol. 169, no. 2, pp. 369–374, 2007.