RESEARCH ARTICLE

Accelerating Agent-Based Computation of Complex Urban Systems

Yu Zou[a], Paul M. Torrens[b,†], Roger G. Ghanem[c] and Ioannis G. Kevrekidis[a,‡]

[a]*Department of Chemical & Biological Engineering and PACM, Princeton University, Princeton, NJ 08544, USA*; [b]*School of Geographical Sciences and Urban Planning, Arizona State University, Tempe, AZ 85287, USA*; [c]*Department of Aerospace and Mechanical Engineering, University of Southern California, Los Angeles, CA 90089, USA*

Despite its popularity, agent-based modeling is limited by serious barriers that constrain its usefulness as an exploratory tool. In particular, there is a paucity of systematic approaches for extracting coarse-grained, system level information as it emerges in direct simulation. This is particularly problematic for agent-based models of complex urban systems in which macroscopic phenomena, such as sprawl, may manifest themselves through emergence from the bottom-up, among diverse agent-actors interacting across scales. Indeed, handling emergence is often crucial in enabling prediction, in supporting decisions, and in facilitating the design, control, and optimization of urban systems. In this paper, we describe and implement a scheme for extracting emergent macroscopic information from local dynamics of agent-based simulation. We compare direct ABM simulation, population-level equation solutions, and *coarse projective integration*. We apply the scheme to the simulation of urban sprawl from local drivers of urbanization, urban growth, and population dynamics. Numerical examples of the three approaches are provided to compare their accuracy and efficiency. We find that our metasimulation scheme can significantly accelerate complex urban simulations, while maintaining faithful representation of the original model.

Keywords: Complexity; Agent-based model; Coarse projective integration; Urban sprawl

†Corresponding author. Email: torrens@geosimulation.com

‡Corresponding author. Email: yannis@princeton.edu

2

1.   Introduction

The rapid uptake in the use of urban simulation since the 1960s (Batty 1976) to explore urban phenomena and to project potential urban futures has resulted in the extension of much urban research to silicon substrates of synthetically-derived cities, particularly for city systems that are difficult to study on the ground (Benenson and Torrens 2004, Batty 2005). This is true for suburban sprawl, a phenomenon that has received widespread recent attention because of concerns regarding the sustainability of sprawling cities (Ewing *et al.* 2003, Song and Knaap 2004, Sultana and Weber 2007) and debate regarding what the causes and consequences of sprawl might be (Torrens 2008). Agent-based models (ABMs) have become increasingly popular as tools for simulating sprawl (Brown *et al.* 2003, Sanders *et al.* 1997, Wu 1998, Parker *et al.* 2003, Batty *et al.* 2007, Loibl *et al.* 2007), because of their flexibility in representing an almost limitless variety of phenomena and systems. Among their advertised advantages, ABM-builders often tout the ability of their tools to simulate complex dynamic systems and related phenomena that defy easy analysis by more traditional forms or academic inquiry or using existing modeling approaches (Batty 2005). Because urban ABMs often rely on fine-grain geographic data, because they scale from local to global geographies, and because their applications delve into the non-linearity of complex coupled systems, the treatment of emergence in simulation is often crucial to their success, particularly when system-level phenomena rely on intra-system agents for their dynamics. Furthermore, urban ABMs often use large numbers of synthetic agents in simulation and treat the details of their many-to-many interactions, and many simulations require frequent reparameterizations and repetitive runs to stabilize. This becomes inefficient for simulations with large numbers of agents and it can be computationally-exhaustive for models with many types of agent behavior, leading to unwieldy and computationally-intensive simulations that require high-performance computing to run (Bandini *et al.* 2001). These difficulties have become manifest at a time when ABMs are being advertised as a planning and decision support tool for many urban applications, specifically because of their ability to trace indelible paths from the microcosm of urban systems to macroscopic outcomes such as sprawl (Torrens 2002). New schemes for (1) handling emergence, and (2) managing the computability of ABMs are required to advance applied urban simulation for these uses. In this paper we introduce methods to improve the ability of ABMs to answer both challenges. We will demonstrate how a metasimulation infrastructure can be built to transform the way that information can be extracted from the "best available", detailed, individual-based computer model that a scientist can write for a given scenario and in the process we will show that the scheme can be used to leverage emergence in simulation for accelerating agent-based computation. We will demonstrate the usefulness of our approach with application to an ABM of suburban sprawl: a phenomenon with particularly thorny complexity.

Our idea is to circumvent the derivation of macroscopic, population-level equations by using fine-scale ABM code as an *experiment* within a larger simulation metasystem. Normally, we would simply run this experiment forward in time and observe its results; indeed, this characterizes much of the existing state of the art (Benenson and Torrens 2004). Instead, we are proposing something different. We wish not just to simulate a model by advancing it in time, but also to interrogate the simulation intelligently, exploiting the premise that some coarse-scale (emergent) system property exists, even if we do not have the macroscopic equations on hand for it in closed form. Essentially, our scheme seeks to address (when a good set of macroscopic, population-level system observables are known) how to design new, informed experiments to obtain information about

system emergence based on specifically-chosen values of those local observables. This is achieved by creating a set of mathematics-assisted computational superstructures that are "wrapped around" a given ABM. The wrapper allows the extraction of broad-scale, system-level information – easier, faster, better – than existing techniques will currently permit. This information may be extracted from the simulations themselves, even when meaningful, coarse-grained, macroscopic variables (observables) of the individual-based ABM are not available *a priori*; indeed, one can study how to build agent-based computational experiments to detect locally-good coarse coordinates that efficiently represent the salient features of a simulation.

While direct agent-based simulation can be feasibly employed to study urban sprawl, the total simulation time in traditional approaches may pose a challenging disadvantage when one needs to investigate the system-level information for many interacting entities of the urban sprawl process; this quite often requires a large Monte-Carlo ensemble of simulation runs to resolve. Of course, high-performance and parallel computing techniques using a Message Passing Interface (MPI) (Gropp *et al.* 1996) or employing many-core Graphics Processing Units (GPUs) (Galvao *et al.* 2008) may be used to enhance computing speed, but it would be useful to also provide an effective, accelerated modeling technique to increase efficiency on the algorithm/software side. Indeed, this paper contains a representative illustration of such an algorithm. Specifically, we will introduce a scheme to *solve* explicit macroscopic equations *without deriving them in closed form*; this is the so-called equation-free approach (Theodoropoulos *et al.* 2000, Kevrekidis *et al.* 2003, 2004), and we will present a particular equation-free algorithm: coarse projective integration. We will contrast this (and the direct ABM simulation) with the simulation of explicit macroscopic equations (with or without the need of some precomputation) when such explicit equations are available. These tasks will be illustrated using an existing ABM (Torrens 2006).

The paper is organized as follows. We discuss related work in Section 2. In Section 3 we provide an overview of the simulation models for urban sprawl that will be used in this work. Explicit equations of agent population profiles for different models are derived in Section 4. Application of coarse projective integration as a numerical acceleration method, to urban sprawl simulation is introduced in Section 5. Numerical experiments comparing solutions from the direct simulation, explicit equations, and coarse projective integration are detailed in Section 6, before drawing conclusions in Section 7.

2.   Related Work

It has traditionally been difficult to capture and measure emergence in complex urban phenomena in anything but abstract terms (for example, discovery that city-systems are fractal (Batty and Longley 1994) or that they are scale-free (Batty 2008)), but the current popularity of using ABMs to model cities has forced the issue, largely because of a common reliance upon emergence in ABMs to associate agent actions and interactions between diverse phenomena and scales (Epstein 1999). Recent approaches to extract complexity signatures from models have provided more concrete metrics of emergence, but the procedures used generally focus on the end-condition of a model (rather than its complete trajectory through simulation) and rely on reasonably well-characterized primitives as a template for matching the model to the real-world, based on self-similarity (Xie 1996), space-filling (Ward *et al.* 2000b), fractal dimension (Batty 2005), spatial autocorrelation (Wu 1998), landscape metrics of state configuration and composition

4

(Herold *et al.* 2005), and fuzzy pattern-matching to known maps or satellite data (Power *et al.* 2000), for example. However, a crucial issue remains unaddressed in these cases: while useful, these techniques are not capable of treating the emergence of novelty.

Other approaches use brute-force schemes to exhaustively sweep the parameter-space (Couclelis 1997) of a model by running simulations for many different permutations and combinations of parameter values (Li and Yeh 2000). This follows efforts in mathematics and computer science to explore the complete space of possibilities for particular classes of automata, in an effort to derive universal laws for various phenomena (Wolfram 2002). The idea is that if such laws were understood, the end-state of a given system could be predicted given any parameterization of the system. City systems are formidably more complicated, messy, complex, and even chaotic than many systems explored in computer science, however, and it is possible that a search for universality is a futile proposition in these cases (Wolfram 1984, Batty and Torrens 2005). Exhaustively sweeping detailed models is time-consuming and can generate massive volumes of results to be evaluated, and so brute-force techniques of this kind usually rely on some averaging procedure, which smoothes results (Clarke *et al.* 1997, Goldstein *et al.* 2004). However, the smoothing could, potentially, mask novelty or subtleties in emergence.

In other approaches, perturbation is used to artificially "wiggle" a model into new trajectories. This can be random, or it can be based on some rules. The SLEUTH model, for example, links changes in model parameters dynamically to evolving growth conditions in simulation (Clarke *et al.* 2007). Self-modifying rules are used to accelerate or dampen growth in a model if particular benchmarks are reached. In effect, this is used as a mechanism to calibrate a model to historical urban scenarios, but it could feasibly be related to emergence. Again, this requires that sensible rules or meaningful benchmarks be available *a priori*. Andersson and colleagues (Andersson *et al.* 2002) developed an interesting variation on this scheme to account for distinct phase transitions in simulation. It is unlikely that cities evolve to neat steady-state conditions in this way, but the approach is particularly innovative. A different strategy is used in geographic models in social science, whereby genetic algorithms are employed to mutate models as they evolve (Epstein and Axtell 1996). Usually, the mutations that bubble forth are then detected using techniques from multivariate statistics (cluster analysis, principal components analysis). This approach relies on requirements for volumes of essentially blind trial-and-error computation and any stopping rules that are invoked tend to be borrowed from genetics (Axelrod 1997) with little consideration of their inconsonance for social science phenomena. Batty and colleagues (Batty *et al.* 1999) have experimented with this approach in simple physics-oriented urban growth models.

Our approach is relatively distinct. Rather than averaging many long model runs or repeating numerous simulations with varying parameters, we run ABMs as local, *short-burst experiments* within a metasimulation framework. This framework makes it possible for us to use many scientific computation algorithms from traditional, continuum numerical analysis to perform tasks such as numerical integration in time, fixed point computations, stability and parametric analysis and more. Accelerated simulation (through coarse projective integration) will be demonstrated here in terms of expected values of the agent population density. Time derivatives of these "coarse variables" are estimated on the fly and repeatedly used to extrapolate them in the time domain, resulting in more efficient computing than the direct (ABM) simulation.

3.   Agent-Based Drivers of Suburban Sprawl

In this section we describe the formulation of our scheme, using as our illustrative example a sprawl ABM developed in Torrens (2006). Note, however, that the procedure is general enough that it could be realistically "wrapped around" any ABM. In our illustrative ABM, the dynamics of sprawl through urbanization, urban growth, and population dynamics are mainly driven by exogenous sources, endogenous change, and mobilization of agents. The implementation of each of these drivers in the simulation will be described in detail in the following subsections.

3.1.   *Exogenous sources*

Exogenous change of the agents is imposed in the urban sprawl simulation at a macrolevel. External agents are additively infused into the city domain typically at some fixed gateways. (These are synonymous with historical portals for immigrants, or existing settlements with some spatial advantage or geographical inertia.) The rate of exogenous change is determined either by user definition or through historical US census data on urban population growth (records of which are available since the eighteenth century). The volume of external agents, $D_n(x, y)$, and the agent population, $P_n(x, y)$, at a discrete time instant t_n determine the population at time t_{n+1}:

$$P_{n+1}(x, y) = P_n(x, y) + D_n(x, y). \tag{1}$$

Here (x, y) is an integer-numbered coordinate on a 2D spatial grid. The population $P_{n+1}(x, y)$ will also be subject to modification due to endogenous change and movement of agents. This effect will be discussed in the following sections.

3.2.   *Endogenous change*

Urban population dynamics involve a balance between migration effects and natural (crude) growth and decline. In the model this is implemented through internal changes such as immigration (im) of population to a land location (x, y), emigration (em), the birth rate (b), and the death rate (d):

$$P_{n+1}(x, y) = P_n(x, y)\delta_n(x, y), \tag{2}$$

where $\delta_n(x, y)$ is a linear combination of internal change rates: $\delta_n(x, y) = im_n(x, y) - em_n(x, y) + b_n(x, y) - d_n(x, y)$.

3.3.   *Agent movement*

The movement of agents (considered as migration of population through residential relocation) can also be viewed as a mapping from time t_n to t_{n+1}, i.e.,

$$S_{n+1} = \Pi(S_n), \tag{3}$$

where S_n is the location of an agent at time t_n in a 2D spatial grid. The operator Π may consist of $m(\geq 1)$ movement steps which follow prescribed (deterministic or stochastic)

rules. The resulting intermediate locations of an agent between t_n and t_{n+1} will then be denoted as S_n (i.e., $S_{n,0}$), $S_{n,1}, S_{n,2}, ..., S_{n,m-1}, S_{n+1}$ (i.e., $S_{n,m}$). In Torrens (2006) five rules for mobilizing agents are suggested which include *leap-frog, immediate, nearby, irregular,* and *road-like* movement. Details of these rules are described below.

3.3.1.   Leap-frog movement

In this movement rule, agents perform a random walk in a Moore neighborhood of size N_m between two successive intermediate locations $S_{n,k}$ and $S_{n,k+1}$ (Fig. 1 (a)). This represents growth by speculative development, ahead of an existing urban mass (Benguigi *et al.* 2001).

3.3.2.   Immediate movement

Between the two time steps t_n and t_{n+1}, an agent randomly chooses a neighboring grid cell to locate itself and then circulates around the (size 1) Moore neighborhood of its original position for $m - 1$ steps (Fig. 1 (b)). This represents growth by accretion (Batty 1991).

3.3.3.   Nearby movement

Here agents circulate around the Moore neighborhood (of size 2) of their original position at t_n. Specifically, they first move around an inner ring (Size 1 Moore neighborhood) as in the immediate movement rule; then, if the number of intermediate steps m is larger than 8, the agent will randomly choose a neighboring cell within a distance of $\sqrt{2}$ in the outer cell ring, and continue moving around this outer ring for $m - 8$ steps (Fig. 1 (c)). This represents intra-urban growth by in-filling (Yeh and Li 2000).

3.3.4.   Irregular movement

The irregular movement rule mobilizes an agent at lattice position $S_{n,k}$ to a cell $S_{n,k+1}$ in the Size 1 Moore neighborhood of $S_{n,k}$. The cell $S_{n,k+1}$ is randomly chosen, so that the distance between $S_{n,k+1}$ and S_n is no less than the distance between $S_{n,k}$ and S_n (Fig. 1 (d)). This represents growth that is constrained by natural features (Ward *et al.* 2000a).

3.3.5.   Road-like movement

For a given grid cell, a path starting at the cell is first "paved" following the irregular rule. At each cell along the path we assign a weight which is inversely proportional to the distance between the cell and the starting cell. All agents located at the starting cell are then deposited at cells along the path using the Roulette Wheel selection algorithm (Goldberg and Deb 1991) (Fig. 1 (e)). In this manner, a cell with a large weight will have on average more agents standing on it than a cell with a small weight. This represents growth along road and other transportation corridors (Hoyt 1964).

The population distribution will dynamically evolve as agents move. Clearly, we can evolve the ABM and *observe* the evolution at a coarse-grained, population (as opposed to individual agent) level; the coarse-graining could involve averaging over spatial neighborhoods, over temporal periods, and –even– over several sample path simulations (for stochastic rules). A vital question then arises: is it possible to derive (based on the detailed, individual-based, rules) a macroscopic equation in terms of the agent number density only? That is, can we derive an equation that, given only the instantaneous agent number density profile, can be used to predict its evolution in the future? One perceives here an analogy between atomistic simulations (e.g., molecular dynamics simulations of
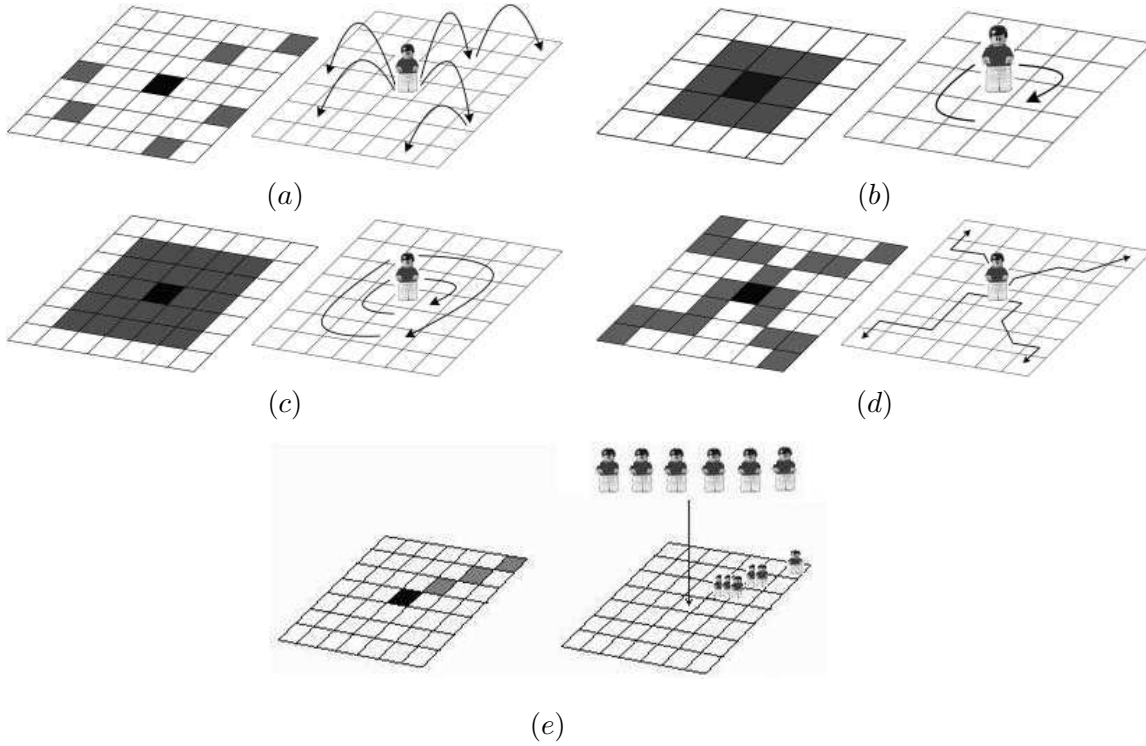
Figure 1. Agent movement rules. (a) Leap-frog rule; (b) Immediate rule; (c) Nearby rule; (d) Irregular rule; (e) Road-like rule.

fluid flow) and traditional continuum numerical models (e.g., the Navier-Stokes equations for the macroscopic evolution of density and momentum fields in a flow). For *interacting agent* simulations, the derivation of accurate continuum, explicit macroscopic equations is the exception rather than the rule. The agents in our illustrative example, however, are *noninteracting*, and one may expect that the ensemble average (over several simulation realizations) of the population profile, $P(x,y)$, over the 2D grid could satisfy an explicit evolution equation that will take into account all the three effects for population change and growth that were just mentioned:

$$P_{n+1}(x,y) = \Phi(P_n(p,q), D_n(p,q), \delta_n(p,q); p \in I[-\infty,\infty], q \in I[-\infty,\infty]), \qquad (4)$$

Here p and q are traversed by the operator Φ over the integer-number domain $I[-\infty,\infty]$ over which the expected profile $P_{n+1}(x,y)$ is to be calculated.

4.   Equations for Agent Populations

Equation-free computational techniques are designed for cases where explicit macroscopic equations *conceptually exist* but are not available in closed form. In some of our illustrative examples such macroscopic, population-level equations *can be* explicitly *analytically* derived; in other cases they can be explicitly derived, but certain terms in their right-hand-side must be obtained *numerically* through pre-computation. Having such population-level equations (whether analytically derivable, or obtained through precomputation) allows us to compare the results of the full ABM simulation to those of the macroscopic equations as well as to those of our equation-free approach, for validation

8

purposes.

A population-level equation Eqn. (4) can indeed be explicitly derived by computing the migrating probability of agents moving into and out of a grid cell. In our case, Eqn. (4) would read

$$P_{n+1}(x,y) = \delta_n(x,y)(P_n(x,y) + D_n(x,y)) - \delta_n(x,y)(P_n(x,y) + D_n(x,y))\eta_o(x,y)$$
$$+ \Sigma_{(p,q)\in M_{N_m}(x,y),(p,q)\neq(x,y)}\delta_n(p,q)(P_n(p,q) + D_n(p,q))\eta_{i,(x,y)}(p,q), \qquad (5)$$

where $M_{N_m}(x,y)$ is the Moore neighborhood of (x,y) with size N_m. $\eta_o(x,y)$ stands for the move-out probability at (x,y) and $\eta_{i,(x,y)}(p,q)$ the move-in probability from (p,q) to (x,y). The difference $\eta_{i,(x,y)}(x,y) = 1 - \eta_o(x,y)$ is the probability that an agent remains in its original position. Eqn. (5) is thus rewritten as follows:

$$P_{n+1}(x,y) = \Sigma_{(p,q)\in M_{N_m}(x,y)}\delta_n(p,q)(P_n(p,q) + D_n(p,q))\eta_{i,(x,y)}(p,q). \qquad (6)$$

If no boundary or barriers occur, $\eta_{i,(x,y)}(p,q)$ is homogeneous with respect to (x,y). Furthermore, $\eta_{i,(x,y)}(p,q) = \eta_{i,(p,q)}(x,y)$ due to symmetry. However, if boundaries or barriers exist in an urban region, $\eta_{i,(x,y)}(p,q)$ will be non-uniform, and it may require prohibitively expensive computation to accurately precompute this migrating probability for every grid point in the simulation domain. In what follows, analytical as well as numerical derivation of the migrating probabilities for the five agent movement rules will be discussed for simulation without boundaries or barriers.

For simple movement rules like *leap-frog, immediate*, and *nearby*, the probabilities $\eta_{i,(x,y)}(p,q)$, $(p,q) \in M_{N_m}(x,y)$ can be analytically obtained. In particular, for the *leap-frog* rule, since an agent has equal probability to move to any cell of its size N_m Moore neighborhood between two intermediate steps (n,k) and $(n,k+1)$, an analytical equation for population profiles is available to link $P_{n,k}(x,y)$ and $P_{n,k+1}(x,y)$, i.e.,

$$P_{n,k+1}(x,y) = \Sigma_{(p,q)\in M_{N_m}(x,y)}\delta_{n,k}(p,q)(P_{n,k}(p,q) + D_{n,k}(p,q))\eta_{i,(x,y)}(p,q), \qquad (7)$$

where $\eta_{i,(x,y)}(x,y) = 0$ and $\eta_{i,(x,y)}(p,q) = 1/((2N_m + 1)^2 - 1)$ for any $(p,q) \in M_{N_m}(x,y)$ and $(p,q) \neq (x,y)$.

In the *immediate* rule, an agent has equal probability to move to any cell of its size 1 Moore neighborhood in m intermediate steps between time t_n and t_{n+1}; thus Eqn. (6) is a valid analytical equation for this rule where $N_m = 1$, $\eta_{i,(x,y)}(x,y) = 0$ and $\eta_{i,(x,y)}(p,q) = 1/8$ for $(p,q) \in M_1(x,y)$ and $(p,q) \neq (x,y)$.

For the *nearby* rule, Eqn. (6) is also valid and $N_m = 2$. As in the immediate rule, the probability $\eta_{i,(x,y)}(x,y) = 0$. $\eta_{i,(x,y)}(p,q)$ for $(p,q) \in M_1(x,y)$ equals 0 as well. Simple reasoning using conditional probability shows that $\eta_{i,(x,y)}(p,q)$ for (p,q) in the outer ring of $M_2(x,y)$ takes one of three values $p_1 = 1/8(2/5 + 1/3)$, $p_2 = 1/8(1/5 + 1/3)$, and $p_3 = 1/40$, showing a pattern like $(p_1, p_2, p_3, p_2, p_1, p_2, p_3, \cdots)$ along the clockwise direction, with the first p_1 corresponding to a corner cell in the outer ring (see Table 1).

For other rules Monte Carlo simulation has to be used to compute, off-line, a look-up table for the probabilities $\eta_{i,(x,y)}(p,q)$ in Eqn. (6); this reduces to computation of only $\eta_{i,(0,0)}(p,q)$ for the spatially homogeneous case. We use ensemble realizations to simulate the positions an agent will move to when starting from $(0,0)$, counting the number of occurrences of position (p,q), and then calculating the probability $\eta_{i,(0,0)}(p,q)$.

p_1	p_2	p_3	p_2	p_1
p_2	0	0	0	p_2
p_3	0	0	0	p_3
p_2	0	0	0	p_2
p_1	p_2	p_3	p_2	p_1

Table 1. Migrating probabilities from cells in the size 2 Moore neighborhood to the central cell for the nearby rule. $p_1 = 1/8(2/5 + 1/3)$, $p_2 = 1/8(1/5 + 1/3)$ and $p_3 = 1/40$.

5.   Coarse Projective Integration for Agent Populations

We use the Coarse Projective Integration (CPI) method of the equation-free framework developed in Theodoropoulos *et al.* (2000), Kevrekidis *et al.* (2003, 2004), to accelerate direct ABM simulation. This approach exploits the smoothness of coarse-level, macroscopic variables in multiscale complex system models, so that these variables can be accurately extrapolated over a large time interval and, accordingly, the overall computation time for the direct, fine-level simulation can be compressed. This approach mimics solving the population-level equations (had they been available to simulate) and is designed for the case when closed equations for the population-level variables are not explicitly available. This is particulary relevant to simulation of complex urban systems, where we may actually be using the simulation as a tool to think with and to quickly and efficiently explore what appropriate rules might be.

Specifically, the method employs a *time-stepper* for population-level variables (described immediately below) to repeatedly (1) obtain successive snapshots of coarse variable evolution; (2) numerically estimate temporal derivatives of these variables; and then (3) extrapolate ("project") the values of the variables to a future time using the estimated temporal derivatives. The time-stepper consists of three essential stages. The first one is a *lifting* procedure that, given the population-level variables' values, generates an ensemble of fine-level-state realizations consistent with these macroscopic values. The second stage is *fine-level simulation*, which evolves the fine-level states for some time interval. The third stage is the calculation (from the result of the ABM simulation realizations) of the coarse-level variables at the final time, which we call *restriction*.

In our case the macroscopic variables consist of a (discretized, possibly on a coarse mesh) population density profile. One way to design the time-stepper for the CPI is to generate realizations of ensemble agents from such a real-numbered agent (population) profile and to let each realization run according to the movement rules for the agents.

For general scenarios of the population profile evolution, we simply chose to extrapolate grid (expected) values of the population profile at every grid point to a future time. The CPI procedure is as follows:

(1) *Lifting*: Generate ensemble realizations of integer-numbered population profiles $P_{n,q}(x,y), q = 1, 2, \cdots, N_e$, where N_e is the ensemble size, consistent with a real-numbered profile $P_n(x,y)$. Then randomly produce an ensemble of agents for each integer-numbered profile $P_{n,q}(x,y)$.

(2) *Fine-level Simulation*: Run each integer-numbered population profile $P_{n,q}(x,y)$ through mobilizing ensemble agents in realizaton q according to the movement rule(s) chosen.

(3) *Restriction*: At some later time steps $t_{n+1}, t_{n+2}, \cdots, t_{n+T}$, calculate the profiles $P_{n+1,q}(x,y)$, $P_{n+2,q}(x,y)$, $\cdots$, $P_{n+T,q}(x,y)$ for each realization by counting the number of agents in each (x,y), and then obtain the ensemble-averaged profiles $P_{n+1}(x,y), P_{n+2}(x,y), \cdots, P_{n+T}(x,y)$.
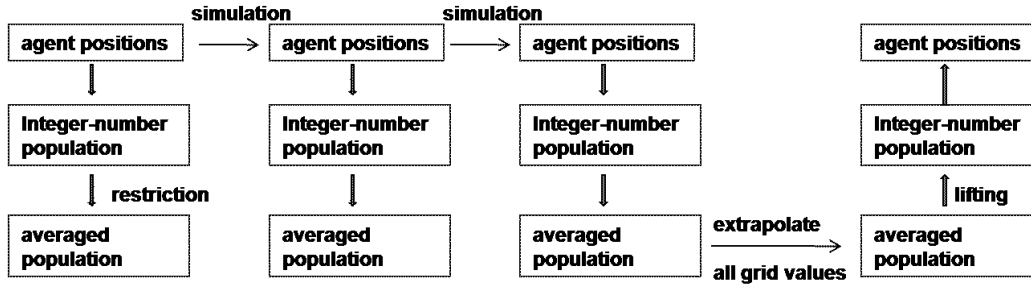
Figure 2. The coarse projective integration scheme applied to urban sprawl.

(4) *Extrapolation*: Numerically estimate the temporal derivative $\frac{dP_{n+T}(x,y)}{dt}$ by implementing a least-square fitting of the last T_f population profiles $P_{n+T-T_f+1}(x,y)$, $P_{n+T-T_f+2}(x,y), P_{n+T}(x,y)$. The extrapolated population at (x,y) is then calculated by

$$P_{n+T+T_e}(x,y) \approx P_{n+T}(x,y) + \frac{dP_{n+T}(x,y)}{dt}T_e, \qquad (8)$$

where T_e is the extrapolation time interval.

The CPI scheme is schematically depicted in Fig. 2.

6.   Numerical Results

Following the original ABM simulation experiment in Torrens (2006), we ran the model for three sprawl scenarios: general growth, polycentric growth, and a site-specific application to the American midwest megalopolis (Chicago and vicinity, around Lake Michigan). To evaluate the performance of the scheme, we performed first direct simulation and then CPI in a parallel computing environment with 25 3.2GHz Xeon processors (part of Princeton's TIGRESS supercomputing cluster). If population-level equations can be derived, it is clear (and we will confirm this below) that simulating the discretized continuum equations is the fastest computational alternative; the migrating probabilities that appear in these equations may be analytically obtainable, or they may require an off-line, precomputation step, which is also implemented in parallel. In the following we will first confirm that explicit equation solutions can be used to accelerate the simulation of three scenarios of urban sprawl as performed in Torrens (2006). We will also show that CPI provides an alternative approach to accelerating these ABM simulations; importantly, this type of acceleration is possible even when explicit (either with or without precomputation) equations are not available. In our illustrative CPI computations the simulation time interval T is set to 5, the time interval for least-square fitting T_f is set to 3, and the extrapolation interval T_e is also set to 5.

6.1.   *General growth scenario*

In this scenario, a central city and two competing cities are located in a 2D domain which consists of 801×601 grid points. Agents may enter the central city through three gateways, while each competing city has only one gateway. Exogenous change rates in the central city are approximately 16,000 per time step and the immigration rates in the two

competing cities are roughly 75% of that in the central city. Heterogeneous endogenous change rates are assigned to grid points in the simulation domain. The leapfrog rule is used to mobilize agents. 50 copies of the population are simulated to obtain the ensemble-averaged population. The computation is implemented up to simulation time $t = 300$. (A unit of time here roughly corresponds to a year in the real world, although, since the simulation is abstract, this is not quantitatively important. In a later scenario, the model will be applied to a real-world analog, and then correlations between simulation and real-world timing can be drawn.) Fig. 3 illustrates a comparison of the ensemble-averaged population computed by the direct ensemble simulation and the explicit equation solution at four time instances. Relative errors between the two methods are within the level of 3%. Population densities computed from CPI also agree well with the direct ensemble simulation (see Fig. 4) and their relative errors do not exceed 10%. Agent populations at two gateways (city centers) computed from the three methods are compared in Fig. 5 which shows excellent visual agreement. In the coarse projective integration, despite slight noise in the extrapolated profile relative to that in direct simulation, the extrapolated profile quickly adjusts back to the true profile evolved by the direct simulation, typically within one time step or two (the "healing period" in equation-free computation).
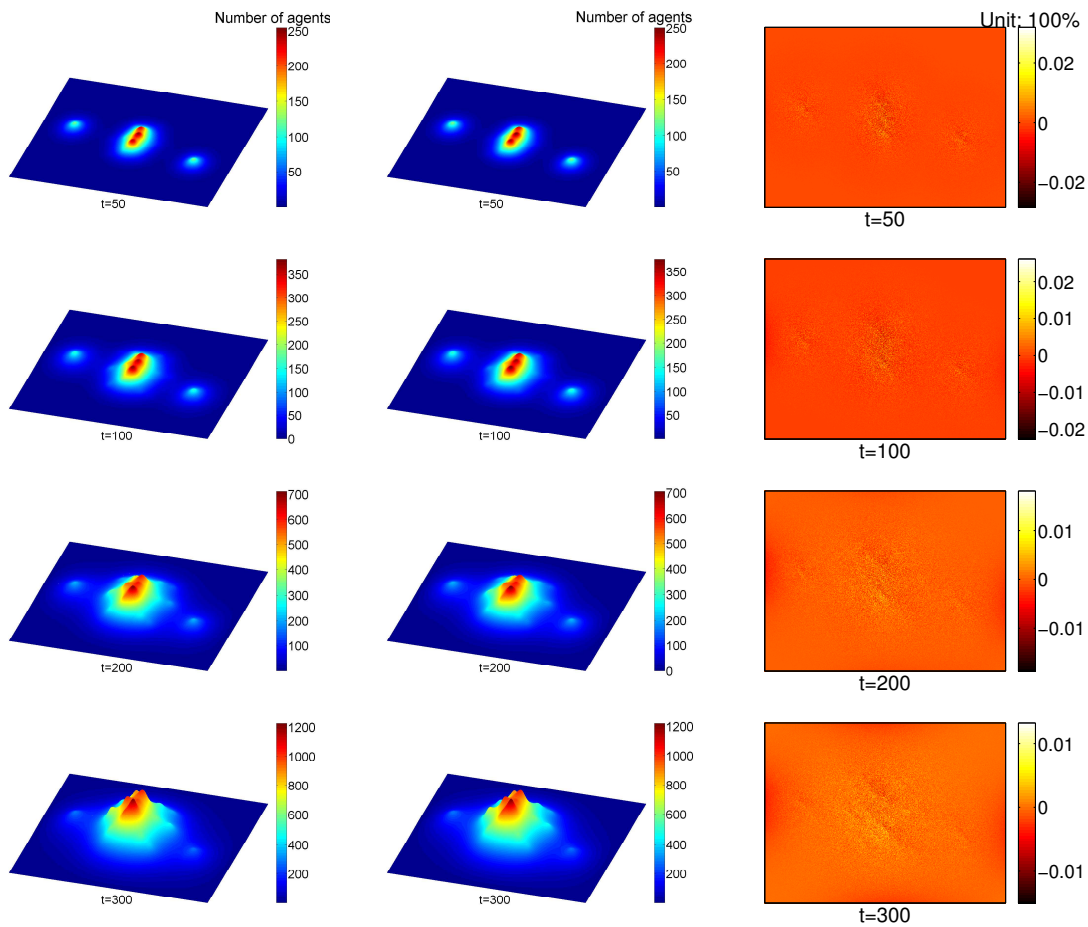

Figure 3. Comparison of ensemble-averaged agent populations computed using the direct ABM simulation and a single explicit equation solution at four time instances. Agent motion follows the leapfrog rule. Left column: population from direct ensemble simulation; Middle column: population from explicit equation solution; Right column: relative errors between explicit equation solution and direct ensemble simulation.
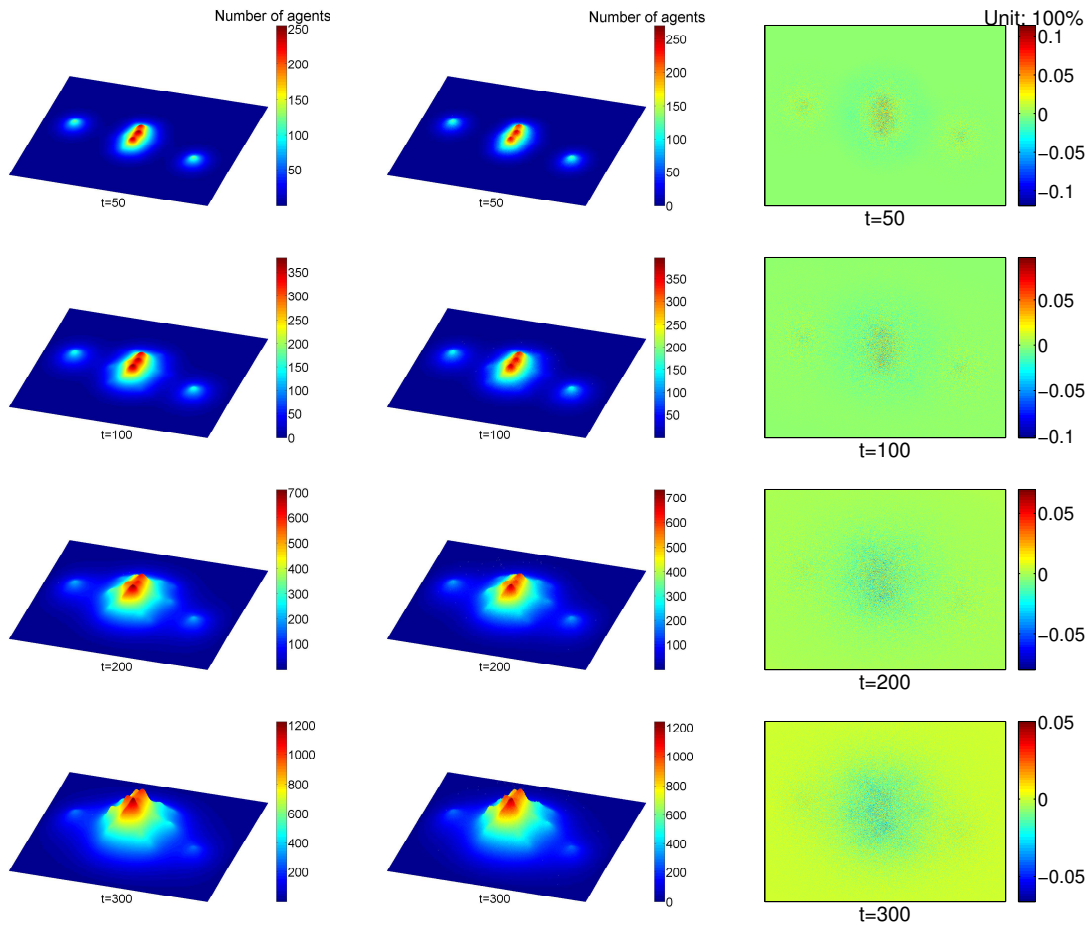
12

Figure 4.   Comparison of ensemble-averaged agent populations computed using the direct ABM simulation and those computed through coarse projective integration at four time instances. Agent motion follows the leapfrog rule. Left column: population from direct ensemble simulation; Middle column: population from coarse projective integration; Right column: relative errors between coarse projective integration and direct ensemble simulation.
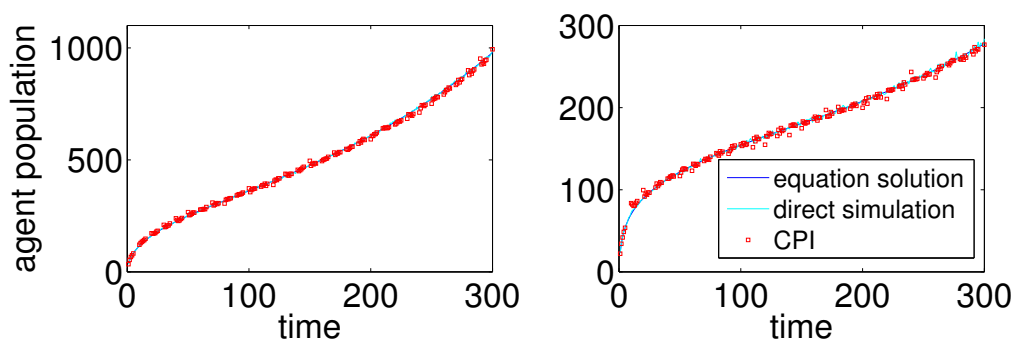

Figure 5.   Comparison of agent population density at two gateways (general growth scenario, see text). Agent motion follows the leapfrog rule. Left figure: the middle gateway of the central city; Right figure: the gateway of the right competing city.

6.2.   *Polycentric growth scenario*

In the second scenario, the urban settings and the agent exogenous change rates are the same as in the first scenario. However, the agent migration will now follow a mixed

combination of leapfrog, road-like, irregular, and nearby movements (denoted as the "mixed rule" from now on) at every time step t_n. The migrating probabilities for the mixed rule, computed using 10^4 realizations each consisting of 10^4 agents, are shown in Fig. 6. In this scenario the computation terminates at time $t = 100$.
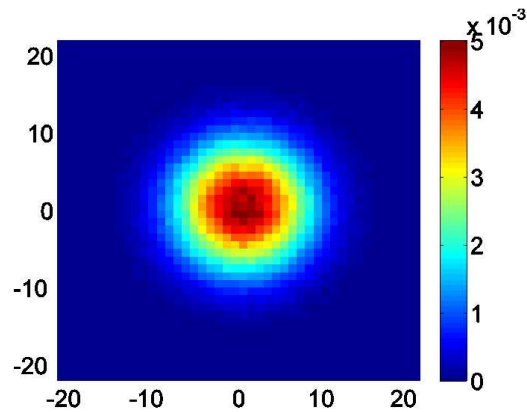

Figure 6.   Migrating probabilities at a representative grid point (at the center of the figure) based on the "mixed rule" (see text).

A comparison of ensemble-averaged populations computed by the three methods is shown in Figures 7 and 8. As in the last scenario, agent populations at two gateways are also compared in Fig. 9. The comparison shows that errors of the equation solution and the CPI relative to direct simulation are within 3% and 10%, respectively.

6.3.   *Midwestern megalopolis scenario*

A realistic history of human population growth around Lake Michigan is simulated in this scenario. Seven seed sites serve as gateways for imposing external change (see Fig. 13 in Torrens (2006)). The external changing rates of human population at the seed sites are derived from historical US census data (see Fig. 14 in Torrens (2006)). As in the previous two scenarios, we assign heterogeneous endogenous change rates to grid points in the 2D simulation domain. Since the migrating probabilities of agents are heterogeneous over the simulation domain, it will require vast computation time to accurately precompute them for each grid point. For this reason, we only use the CPI to accelerate the direct simulation, and compare their results.

The computation is performed up to $t = 200$ simulation time steps (200 years in this case). Fig. 10 shows snapshots of averaged agent populations at $t = 50, 100, 200$ years for both direct simulation and for CPI. Their relative differences are also calculated for these time instances. Due to the error induced by extrapolation at $t = 100$ and 150 years, the maximum relative difference at these time snapshots is (instantaneously) large and close to 50%. Yet this error "heals" quickly (as the figure shows), and at other time instances the relative differences are normally within 30%. The agent population densities at four of the gateways (city centers) are also compared for the two simulation methods in Fig. 11, which shows a close agreement between the direct simulation and CPI.
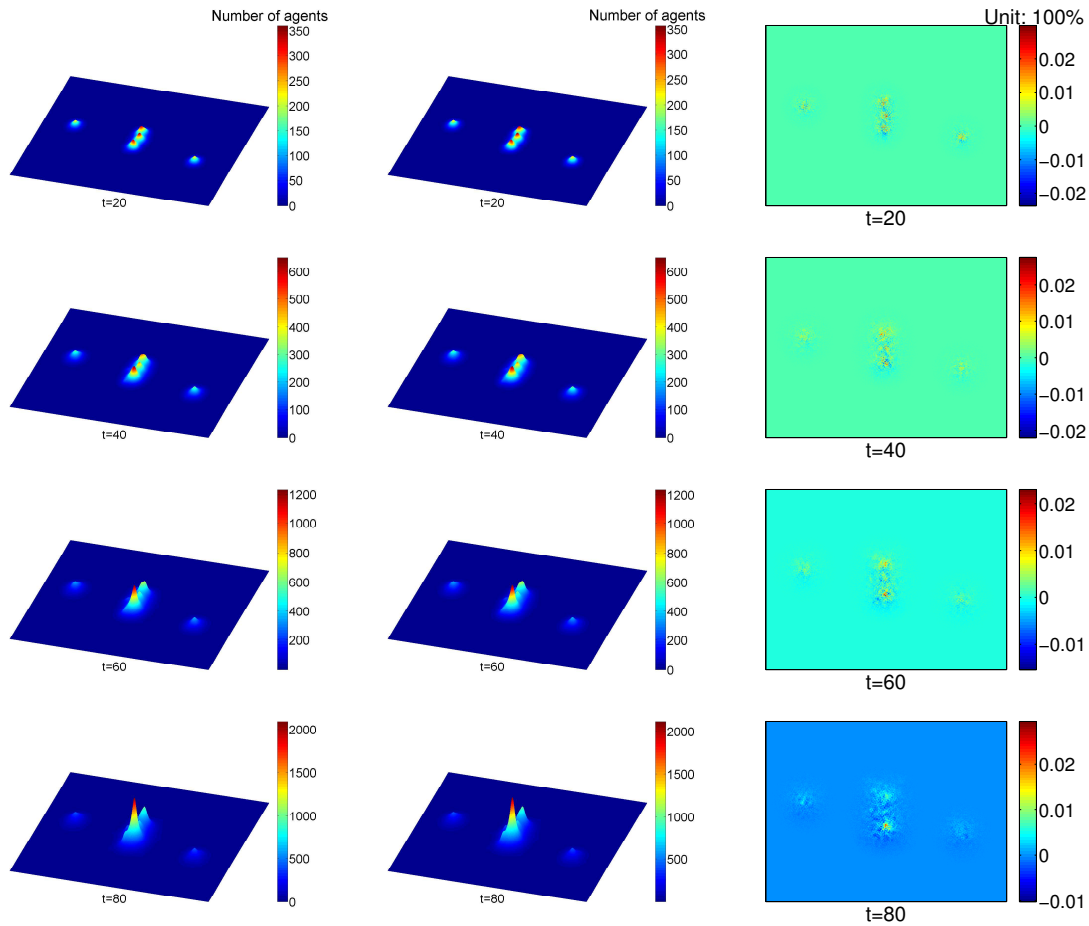
14

Figure 7.   Comparison of ensemble-averaged agent populations computed using the direct ABM simulation and by solving the explicit equation at four time instances. The mixed rule is used to simulate agent motion. Left column: population from direct ensemble simulation; Middle column: population from explicit equation solution; Right column: relative errors between explicit equation solution and direct ensemble simulation.

6.4.   *Computing time*

The computing times for implementing direct simulation, explicit equation solution as well as CPI for the above three scenarios are compared in Table 2.

In the second scenario, the off-line computation for the look-up migrating probability table (necessary for the population-level equation solution) takes only 3 sec for the mixed rule. This table clearly confirms that when population-level equations can be derived (whether analytically or through precomputation) it makes eminent sense to use them in the model simulation instead of the ABM; we expect that such equations will also perform better than CPI. We also know, however, that the ABM themselves are designed and used to capture and understand phenomena not easily describable through explicit coarse-grained, population-level equations. In such a case, CPI would be our only acceleration alternative to the ABM. Relative time savings of the CPI method (compared to the direct ABM simulation) are also shown in the table, indicating that CPI can indeed reduce the simulation time significantly.
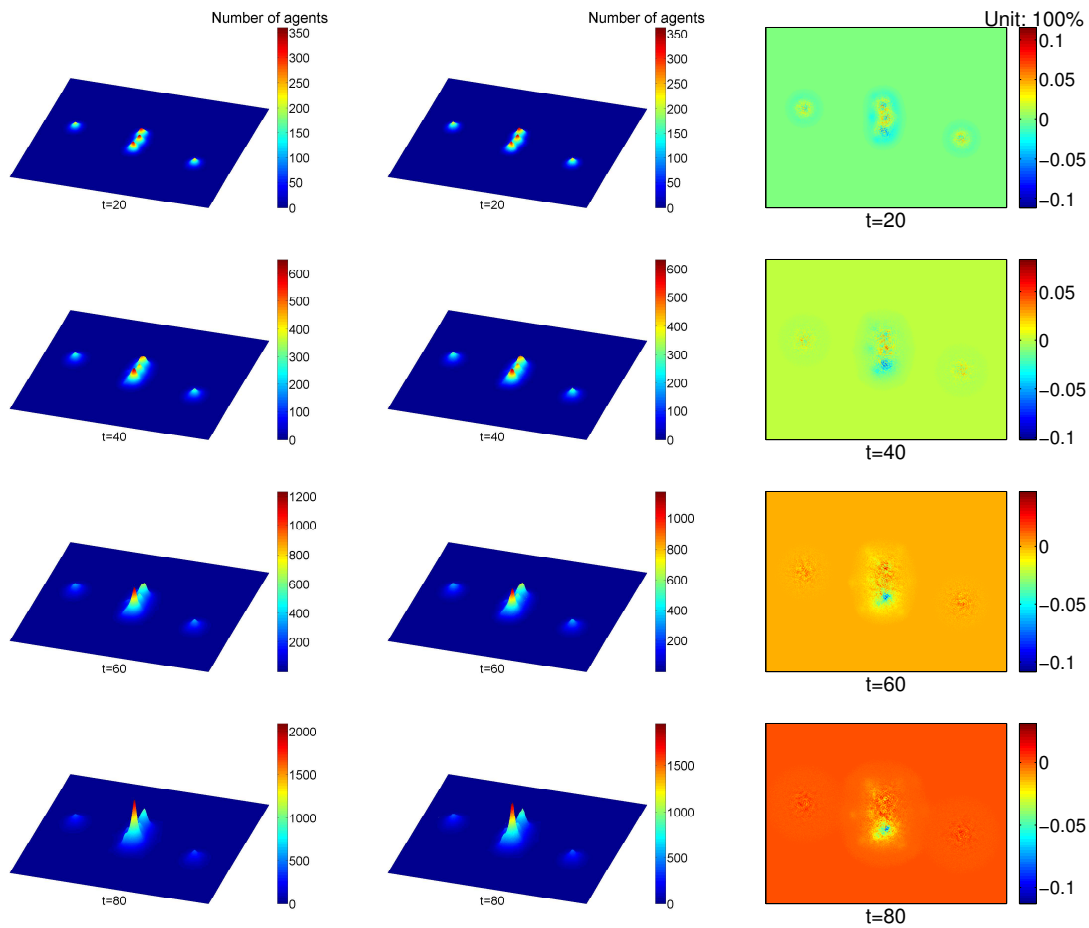
Figure 8. Comparison of ensemble-averaged agent populations computed using the direct ABM simulation and those computed using coarse projective integration at four time instances. The mixed rule is used to simulate agent motion. Left column: population from direct ensemble simulation; Middle column: population from coarse projective integration; Right column: relative errors between coarse projective integration and direct ensemble simulation.
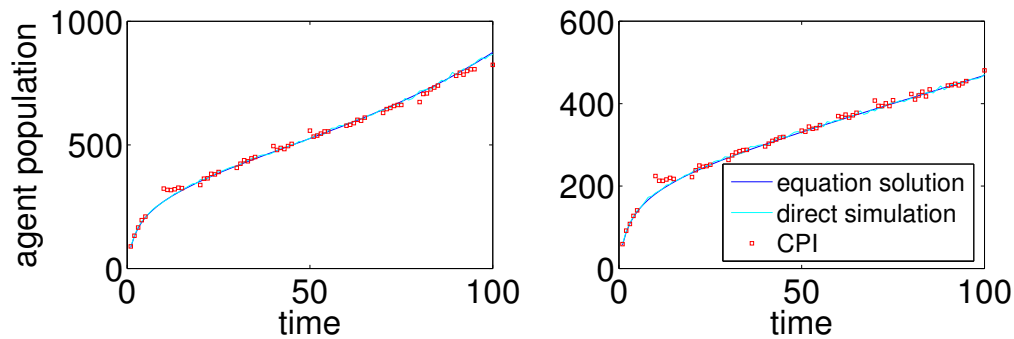

Figure 9. Comparison of agent population density at two gateways (city centers) (polycentric growth scenario, see text). The mixed rule is used to simulate agent motion. Left figure: population evolution at the middle gateway of the central city; Right figure: population evolution at the gateway of the right competing city.

7.   Conclusions

The goal of this work is to introduce a multi-scale computational approach that can accelerate agent-based simulation of complex urban systems. The approach (coarse projective
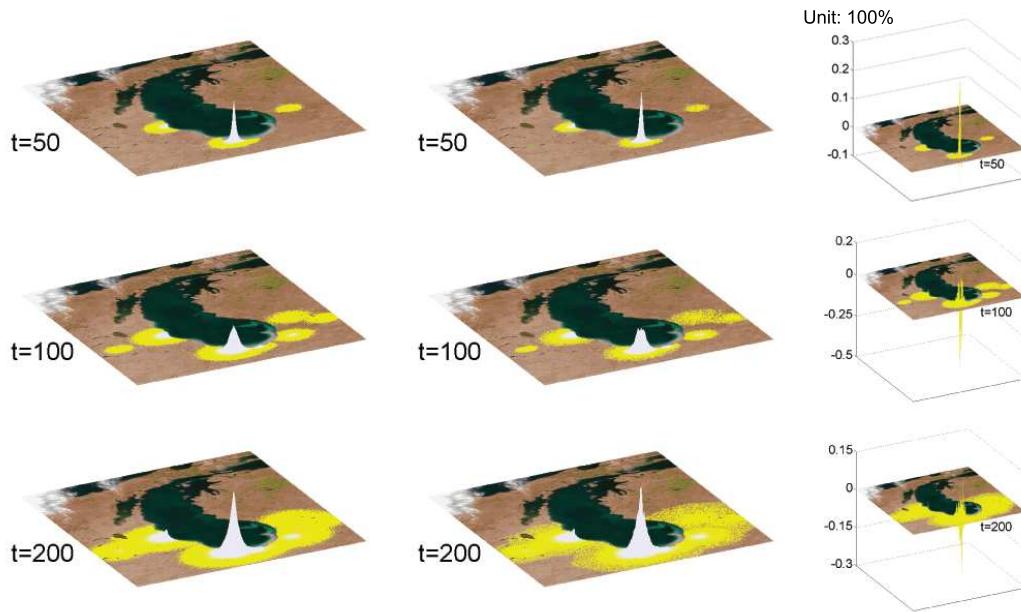
16

Figure 10.   Comparison of averaged population growth around Lake Michigan. Left column: direct ensemble simulation; Middle column: coarse projective integration; Light yellow/white spikes denote populations that are larger than yellow areas. Right column: relative errors between coarse projective integration and direct ensemble simulation.
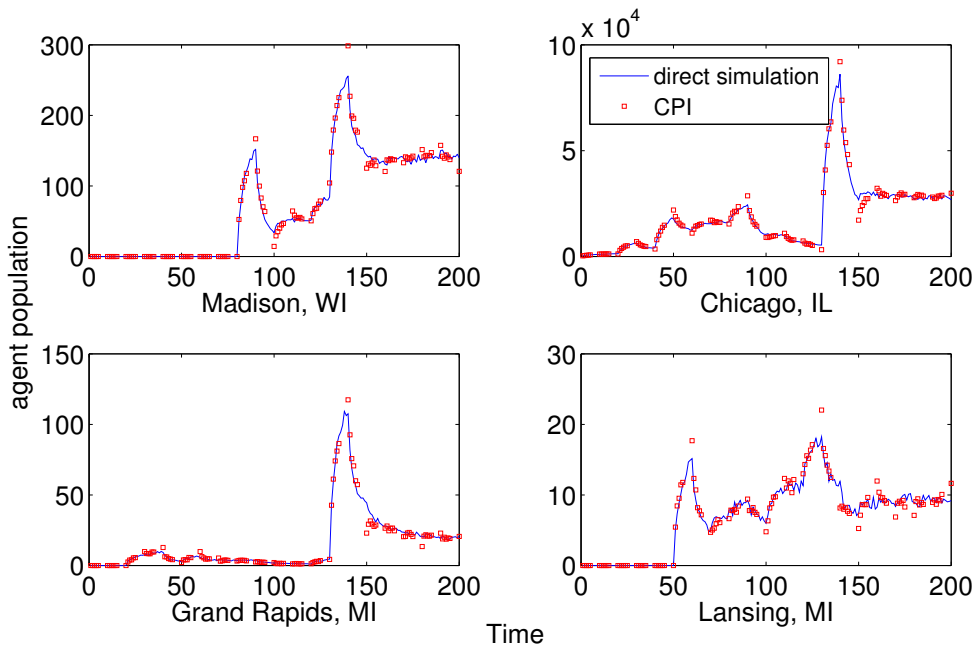

Figure 11.   Comparison of averaged populations, obtained from direct ensemble simulation and from coarse projective integration (CPI), at four of the gateways.

integration) is part of the so-called equation-free framework, that allows one to perform coarse-grained, system-level computations through "wrapper" algorithms – algorithms that design and execute brief bursts of appropriately initialized agent-based simulations.

Scenario type	Direct simulation	CPI	Time savings of CPI relative to direct simulation	Explicit equation	
				Equation solution	Migrating probability
General growth	19356sec	5694sec	70.6%	488sec	NA
Polycentric growth	1788sec	818sec	54.3%	114sec	3sec
Midwestern megalopolis	806sec	470sec	41.7%	NA	NA

Table 2. Computing time of the three methods for simulation scenarios.

This approach is useful when population-level equations *in principle* exist, but cannot be easily derived in closed form.

We also showed here that, if such population-level equations *can* be derived, whether analytically, or with the help of some precomputation, it makes eminent sense to use these equations directly, instead of either direct ABM or equation-free simulations. We chose illustrative examples simple enough that such equations could be obtained, for demonstration purposes; the ambition of the equation-free method, however (and, for that matter, of complex urban ABM simulations) is to explore and understand phenomena for which no useful coarse-grained equations are known at the population level. We compared here results of the three approaches (ABM, equation-based, equation-free) for two movement rules. The results show promise for the usefulness of coarse projective integration methods in accelerating agent-based computation of complex urban phenomena such as suburban sprawl while maintaining a faithful representation of the original ABM.

Our approach promises to dramatically enhance the inherent abilities of agent-based modeling (particularly in their suitability for modeling complex phenomena) by fundamentally improving the efficiency of agent-based modeling and by assisting the systematic extraction of information from such models. Through the choice of appropriate macroscopic observables (the right variables for the coarse-level description) our scheme allows for the principle of emergence in complex systems, and exploits this emergence by deriving alternative, efficient computational schemes. It is important to state that this may be possible even in the absence of *a priori* templates to define what the signatures of that emergence might be. In our examples the macroscopic variable was the "obvious" one: population density. For more complex problems, where correlations between the agents or heterogeneous agent properties are included, modern data-mining tools (like diffusion maps (Coifman *et al.* 2005, Coifman and Lafon 2006)) can be used to find such observables from the simulation itself. This is potentially quite valuable in urban simulation, where ABMs may be used to actually explore what the rules of the system might be or how they might interact over space, time, and domains for varying plans, policies, or assumptions. Our approach may also lead to significant efficiency gains in the computation required to run direct simulations, by replacing blind trial-and-error scenario-sampling and averaging schemes through intelligent protocols for querying models through meta-computation.

There are broader potential advantages to the work that we have presented. It could help reconcile agent-based modeling into a larger "ecology" of mathematical modeling and computation, by transforming the ways in which agent-based models can be used as "black box" experiments. Equation-free computation enables agent-based modeling

to connect to scientific computing algorithms (like fixed point computation, stability, parametric analysis and more) used in the physical sciences and engineering. By linking mathematics-based scientific computation to agent-based modeling, this potentially extends the reach of conventional mathematics-assisted modeling beyond the formal approaches with which it is now associated. Our work also relates to challenges in high-performance computing, particularly as regards the computability of automata models with large numbers of automata and large numbers of rules (Asanovic *et al.* 2006). Indeed, we have illustrated how the agent-based model can be used in the form of rapid-burst computational experiments within a metacomputational framework; in this framework brief ABM simulations are designed, performed, and their results processed to extract information in a fast, quantitative way. The ABM simulator may be changed to include many-to-many interacting agent models with richly-specified rules or behaviors; yet the outer "wrapper" algorithm structure of the metacomputation remains the same. The ambition of this approach is to "liberate" ABM from scenario-collection, and endow it with the degree of quantitation, error control and arsenal of algorithms available for continuum physical/engineering models. To conclude we note that no amount of sophistication in the computation can correct bad modeling: whether the results are obtained fast, or slow, they can be at best as good as the underlying ABM. If the interactions embodied in the ABM rules are physically wrong, our approach cannot correct them – it will (quickly and efficiently) extract the macroscopic consequences of these errors in the macroscopic model predictions.

Acknowledgement

References

Andersson, C., Steen, R., and White, R., 2002. Urban settlement transitions. *Environment and Planning B: Planning & Design*, 29, 841–865.

Asanovic, K., *et al.*, 2006. *The landscape of parallel computing research: A view from Berkeley.* Technical Report UCB/EECS-2006-183, Department of Electrical Engineering and Computer Science, University of California, Berkeley.

Axelrod, R., 1997. *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration.* Princeton, NJ: Princeton Universiy Press.

Bandini, S., Mauri, G., and Serra, R., 2001. Cellular automata: From a theoretical parallel computational model to its application to complex systems. *Parallel Computing*, 27, 539–553.

Batty, M., 1991. Generating urban forms from diffusive growth. *Environment and Planning A*, 23, 511–544.

Batty, M., 1976. *Urban Modelling: Algorithims, Calibrations, Predictions.* Cambridge Urban and Architectural Studies, London: Cambridge University Press.

Batty, M., 2005. *Cities and Complexity: Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals.* Cambridge, MA: The MIT Press.

Batty, M., 2008. The size, scale, and shape of cities. *Science*, 319 (5864), 769–771.

Batty, M. and Longley, P., 1994. *Fractal Cities.* London: Academic Press.

Batty, M. and Torrens, P.M., 2005. Modeling and prediction in a complex world. *Futures*, 37 (7), 745–766.

Batty, M., Xie, Y., and Sun, Z., 1999. Modeling urban dynamics through GIS-based cellular automata. *Computers, Environment and Urban Systems*, 23 (3), 205–233.

Batty, M., Xie, Y., and Zhao, K., 2007. Simulating emergent urban form using agent-based modeling: Desakota in the Suzhou-Wuxian region in China. *Annals of the Association of American Geographers*, 97 (3), 477–495.

Benenson, I. and Torrens, P.M., 2004. *Geosimulation: Automata-Based Modeling of Urban Phenomena*. London: John Wiley & Sons.

Benguigi, L., Czamanski, D., and Marinov, M., 2001. City growth as a leap-frogging process: An application to the Tel-Aviv metropolis. *Urban Studies*, 38 (10), 1819–1839.

Brown, D.G., *et al.*, 2003. Agent-based and analytical modeling to evaluate the effectiveness of greenbelts. *Environmental Modelling and Software*, 19 (12), 1097–1109.

Clarke, K.C., *et al.*, 2007. A decade of SLEUTHing: Lessons learned from applications of a cellular automaton land use change model. *In*: P. Fisher, ed. *Classics in IJGIS: Twenty years of the International Journal of Geographical Information Science and Systems*. Boca Raton, FL: CRC Press, 413–427.

Clarke, K.C., Hoppen, S., and Gaydos, L., 1997. A self-modifying cellular automaton model of historical urbanization in the San Francisco Bay area. *Environment and Planning B*, 24, 247–261.

Coifman, R.R. and Lafon, S., 2006. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21, 5–30.

Coifman, R.R., *et al.*, 2005. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences*, 102, 7426–7431.

Couclelis, H., 1997. From cellular automata to urban models: New principles for model development and implementation. *Environment and Planning B*, 24, 165–174.

Epstein, J.M., 1999. Agent-based computational models and generative social science. *Complexity*, 4 (5), 41–60.

Epstein, J.M. and Axtell, R., 1996. *Growing Artificial Societies from the Bottom Up*. Cambridge, MA: MIT Press.

Ewing, R., *et al.*, 2003. Relationship between urban sprawl and physical activity, obesity, and morbidity. *American Journal of Health Promotion*, 18, 47–57.

Galvao, R., Laycock, R.G., and Day, A.M., 2008. GPU techniques for creating visually diverse crowds in real-time. *In: Proceedings of the 2008 ACM symposium on virtual reality software and technology*, October 27-29. Bordeaux, France: ACM, 79–86.

Goldberg, D.E. and Deb, K., 1991. A comparative analysis of selection schemes used in genetic algorithms. *In*: G.J. Rawlins, ed. *Foundation of Genetic Algorithms*. San Mateo, CA: Morgan Kauffman Publishers, 69–93.

Goldstein, N.C., Candau, J.T., and Clarke, K.C., 2004. Approaches to simulating the "March of Bricks And Mortar". *Computers, Environment and Urban Systems*, 28 (1/2), 125–147.

Gropp, W., *et al.*, 1996. A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing*, 22 (6), 789–828.

Herold, M., Couclelis, H., and Clarke, K.C., 2005. The role of spatial metrics in the analysis and modeling of urban land use change. *Computers, Environment and Urban Systems*, 29, 369–399.

Hoyt, H., 1964. Recent distortions of the classical models of urban structure. *Land Economics*, 40 (2), 199–212.

Kevrekidis, I.G., Gear, C.W., and Hummer, G., 2004. Equation-free: The computer-assisted analysis of complex, multiscale systems. *AIChE Journal*, 50 (7), 1346–1354.

Kevrekidis, I.G., *et al.*, 2003. Equation-free coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level tasks. *Communications in Mathematical Sciences*, 1 (4), 715–762.

Li, X. and Yeh, A.G.O., 2000. Modelling sustainable urban development by the integration of constrained cellular automata and GIS. *International Journal of Geographical Information Science*, 14 (2), 131–152.

Loibl, W., *et al.*, 2007. Simulation of polycentric urban growth dynamics through agents. *In*: E. Koomen, J. Stillwell, A. Bakema and H. Scholten, eds. *Modelling Land-use Change, Progress and Applications*. Dordrecht: Springer, 219–235.

Parker, D.C., *et al.*, 2003. Multi-Agent System models for the simulation of land-use and land-cover change: A review. *Annals of the Association of American Geographers*, 93 (2), 314–337.

Power, C., Simms, A., and White, R., 2000. Hierarchical fuzzy pattern matching for the regional comparison of land use maps. *International Journal of Geographical Information Systems*, 15 (1), 77–100.

Sanders, L., *et al.*, 1997. SIMPOP: A multi-agent system for the study of urbanism. *Environment and Planning B*, 24, 287–305.

Song, Y. and Knaap, G.J., 2004. Measuring urban form: Is Portland winning the war on sprawl?. *Journal of the American Planning Association*, 70 (2), 210–225.

Sultana, S. and Weber, J., 2007. Journey-to-work patterns in the age of sprawl: Evidence from two midsize southern metropolitan areas. *The Professional Geographer*, 59, 193–208.

Theodoropoulos, K., Qian, Y.H., and Kevrekidis, I.G., 2000. 'Coarse' stability and bifurcation analysis using time-steppers: A reaction diffusion example. *Proceedings of the National Academy of Sciences*, 97 (18), 9840–9843.

Torrens, P.M., 2002. Cellular automata and multi-agent systems as planning support tools. Advances in Spatial Science XII, *In*: S. Geertman and J. Stillwell, eds. *Planning Support Systems in Practice*. London: Springer-Verlag, 205–222.

Torrens, P.M., 2006. Simulating sprawl. *Annals of the Association of American Geographers*, 96 (2), 248–275.

Torrens, P.M., 2008. A toolkit for measuring sprawl. *Applied Spatial Analysis and Policy*, 1 (1), 5–36.

Ward, D.P., Murray, A.T., and Phinn, S.R., 2000a. A stochastically constrained cellular model of urban growth. *Computers, Environment and Urban Systems*, 24, 539–558.

Ward, D.P., Phinn, S.R., and Murray, A.T., 2000b. Monitoring growth in rapidly urbanizing areas using remotely sensed data. *The Professional Geographer*, 52 (3).

Wolfram, S., 1984. Universality and complexity in cellular automata. *Physica D*, 10, 1–35.

Wolfram, S., 2002. *A New Kind of Science*. Champaign, IL: Wolfram Media, Inc.

Wu, F., 1998. An experiment on the generic polycentricity of urban growth in a cellular automatic city. *Environment and Planning B*, 25, 731–752.

Xie, Y., 1996. A generalized model for cellular urban dynamics. *Geographical Analysis*, 28 (4), 350–373.

Yeh, A.G.O. and Li, X., 2000. Simulation of compact cities based on the integration of cellular automata and GIS. *In*: S. Bandini and T. Worsch, eds. *Theoretical and Practical Issues on Cellular Automata*. London: Springer-Verlag, 170–178.