Efficient Computation of Emergent Equilibrium in Agent-Based Simulation

Zehong Hu[1,2], Meng Sha[2], Moath Jarrah[1,2], Jie Zhang[2], Hui Xi[3]
[1]Rolls-Royce@NTU Corporate Lab, Nanyang Technological University, Singapore
[2]School of Computer Engineering, Nanyang Technological University, Singapore
[3]Rolls-Royce Singapore Pte Ltd, Singapore

Abstract

In agent-based simulation, emergent equilibrium describes the macroscopic steady states of agents' interactions. While the state of individual agents might be changing, the collective behavior pattern remains the same in macroscopic equilibrium states. Traditionally, these emergent equilibriums are calculated using Monte Carlo methods. However, these methods require thousands of repeated simulation runs, which are extremely time-consuming. In this paper, we propose a novel three-layer framework to efficiently compute emergent equilibriums. The framework consists of a macro-level pseudo-arclength equilibrium solver (PAES), a micro-level simulator (MLS) and a macro-micro bridge (MMB). It can adaptively explore parameter space and recursively compute equilibrium states using the predictor-corrector scheme. We apply the framework to the popular opinion dynamics and labour market models. The experimental results show that our framework outperformed Monte Carlo experiments in terms of computation efficiency while maintaining the accuracy.

Introduction

Agent-based simulation (ABS) has become one of the dominant methods for simulating complex systems with interacting agents (Picault and Mathieu 2011). Emergent ABS systems are a class of ABS models where the evolution of their agents leads to a certain macroscopic steady state. This steady state, termed emergent equilibrium, exists where the aggregated behaviors are maintained despite time-varying states of individual agents. The past few decades have seen the development of many ABS models (Bert et al. 2014; Teose et al. 2011) and powerful simulation platforms (Allan 2010). However, very limited work has been done in efficient computing emergent behaviors in complex systems, including the emergent equilibrium.

To search for the emergent equilibrium of ABS models, one of the most widely used approaches is to conduct Monte Carlo experiments. This approach usually consists of four steps: selecting different parameter values, setting up many initial conditions for each parameter, creating hundreds of ensemble realizations for each initial condition, and executing detailed simulation for each realization long enough to investigate collective behaviours (Tsoumanis et al. 2010). This method is very time-consuming as the simulation of each realization has to start from the very beginning in every iteration. The same problem has also appeared in simulating dynamic systems based on partial differential equations, and it has been solved using the numerical continuation (NC) theory (Allgower and Georg 2003). In the NC theory, when a parameter value changes, new steady states are predicted based on previous ones. Afterwards, the predicted states are corrected through Newton iterations, which avoids recomputing the whole simulation from the beginning to generate the new steady states (Chan and Keller 1982; Shroff and Keller 1993). In addition, the pseudo-arclength method of the NC theory can adaptively select a suitable parameter step size for fast-changing and slow-changing solutions (Chan and Keller 1982). Nevertheless, it is very difficult to apply the NC theory to capture the emergent equilibrium in ABS for two reasons. Firstly, there are no closed-form equations available in ABS models to depict the macroscopic emergent dynamics. The refined simulation is conducted at the micro-level while the numerical computation is conducted at the macro-level. Secondly, the Newton iteration used for correcting the prediction in the NC theory can be of a low efficiency because ABS is highly complex and stochastic which results in great difficulty to eliminate the prediction error with numeric iterations.

Thus, in this paper, we propose a novel three-layer framework aiming to achieve efficient computation of emergent equilibrium in ABS. To our best knowledge, this represents the first systematic effort to apply NC theory to ABS. The proposed framework consists of: 1) a macro-level recursive equilibrium solver with the predictor-corrector scheme to adaptively explore the system parameter space, 2) a micro-level simulator to control the execution of ABS, and 3) a bridge to connect the simulator with the numerical solver and translate macroscopic evolution dynamics into approximated short bursts of suitably initialized microscopic simulation. The macro-level recursive equilibrium solver utilizes the previously obtained equilibrium points to predict the new equilibrium at first. Then, the corrector controls the micro-level simulator through the proposed bridge to obtain short bursts of the microscopic simulation and correct the prediction. As a result, computing the emergent equilibrium from the beginning in every iteration is avoided and the parameter is adaptively adjusted, through which we can significantly enhance the computation efficiency. The framework has been evaluated by two test beds: opinion dynamics and labour market models. The experimental results show that our framework can significantly decrease the computation time required and meanwhile maintaining the accuracy.

Related Work

Previous research works have been exploring different ways to improve the computational efficiency of MCE in obtaining equilibrium. One of the most widely employed approach is to use evolutionary optimization methods to search parameter space and consequently improve the computational efficiency (Stonedahl and Wilensky 2010; Olaru and Purchase 2014). Hence, this approach can enhance the efficiency of MCE through skipping some unuseful points and targetedly selecting some parameter values. As an example, genetic algorithms can be used when the maximum deviation in the neighbourhood is computed for sensitivity analysis. However, such optimization-based methods require well-defined objective functions, and massive simulations are still required to obtain the emergent equilibrium.

Closely related to our work is the equation-free framework originally developed for physicochemistry simulation. It approximates macroscopic evolution dynamics through using short bursts of microscopic simulation (Kevrekidis and Samaey 2009). In physicochemistry applications, the micro-scale simulation usually involves an ensemble of thousands of particles. Transforming from the micro-level to the macro-level (TI2A) typically computes the first few moments of the particles' distribution (e.g. density, momentum, and energy). Transforming from the macro-level to the micro-level (TA2I) uses theoretical distributions to generate particles' states. Compared with the physicochemistry simulation, ABS is more complex and usually needs to handle various dynamics of social, economical or biological systems. The macroscopic distributions of agents' states can be very difficult to be mathematically depicted. Thus, the transformation between micro- and macro-levels for ABS should not be limited to the moments of distributions.

Recently, the equation-free framework has been implemented to analyze collective behaviors in ABS, including information propagation (Tsoumanis et al. 2010), financial market (Liu 2013) and *Escherichia coli* locomotion (Siettos 2014). An agent in these models is represented by only one state variable denoted by a real number. For example, in (Liu 2013), each participant (agent) of a financial market only has a preference attribute which is a number between -1 and 1. Thus, for these one-dimensional homogeneous models, the macroscopic distribution was approximated by a piece-wise linear polynomial or a set of orthogonal polynomials. TI2A was achieved through counting the histogram and TA2I was conducted using the inverse cumulative distribution function. However, these methods are not applicable for multi-dimensional ABS systems with heterogeneous agents. Therefore, a new framework that addresses the limitations of existing methods is proposed in this paper. The contributions of this work include: 1) providing a solution to perform TI2A and TA2I for non-standard and multi-dimensional macroscopic distributions, 2) providing a general framework that can be utilized for both homogeneous and heterogeneous complex ABSs, and 3) combine Heun's method with Newton-GMRES solver to resolve the computation difficulty caused by the stochastic essence of ABS.

Formalization of ABS

North and Macal (2007) argued that agents' attributes should capture both static and dynamic parts. Thus, in this paper, we use $\langle att, s\rangle$ to represent the states of an agent, where att stands for agent's static attributes, s denotes the dynamic states. Furthermore, there are usually several types of agents in ABS. Agents that belong to the same type share the same structure. Hence, given agents' types $\Gamma = \{1, 2, \dots, M\}$ and a set of agents for the ith type $\Sigma_i = \{1, 2, \dots, n_i\}$, the state of an ABS system can be denoted by the vector

$$\boldsymbol{x} = \left[(att^i_j, s^i_j)^{i\in\Gamma}_{j\in\Sigma_i}, s_{Env}\right] \quad (1)$$

where att^i_j and s^i_j stand for the static attributes and dynamic states of the jth agent in the ith type, respectively. s_{Env} represents the environment states. Then, the microscopic evolution of ABS can be represented by $\boldsymbol{x}(t + \mathrm{d}t) = \sigma(\boldsymbol{x}(t), \lambda)$, where $\mathrm{d}t$ is the time step of the micro-scale simulation and λ denotes system parameters.

For the macroscopic variables of ABS, we can denote them by $\boldsymbol{X}$. When conducting microscopic simulation, we can observe a macro-level evolution for $\boldsymbol{X}$ as

$$\boldsymbol{X}(t + \delta t) = \mathfrak{S}(\boldsymbol{X}(t), \delta t, \lambda) \quad (2)$$

where δt denotes the time step of the macro-scale evolution. After a long-time simulation, ABS may reach a certain macroscopic steady state where $\boldsymbol{X}$ does not change any more while $\boldsymbol{x}$ keeps changing. This kind of macroscopic steady state is called emergent equilibrium in this paper. Thus, an ideal emergent equilibrium $\boldsymbol{X}^E$ should fulfill the following equation:

$$\boldsymbol{X}^E = \mathfrak{S}\left(\boldsymbol{X}^E, \delta t, \lambda\right) \quad (3)$$

For different ABS models, the macroscopic variables $\boldsymbol{X}$ used to represent the emergent equilibrium may be different. In this paper, we only focus on a commonly used form, namely, the probability distributions of agents' attributes and states. For simplicity, we assume that all agents that belong to a specific type follow one joint distribution. In some cases, it may not be necessary to consider all microscopic states when calculating the macroscopic evolution. The reason is that some of the states may change very fast compared with the overall system evolution; and hence they can be ignored for the long-term macroscopic computation (Kevrekidis and Samaey 2009). Furthermore, some static attributes do not have high impact on the macroscopic collective behavior and can also be ignored. Properly choosing the states and attributes can lower the dimension of macroscopic variables and thus significantly reduce the computational difficulty. Nevertheless, the simplification of macroscopic variables may lead to the error in equilibrium states, which should be considered for further analysis. Therefore, the macroscopic variable can be written as

$$\boldsymbol{X} = \left[P^1(Att^1, S^1), \dots, P^M(Att^M, S^M), S_{Env}\right] \quad (4)$$

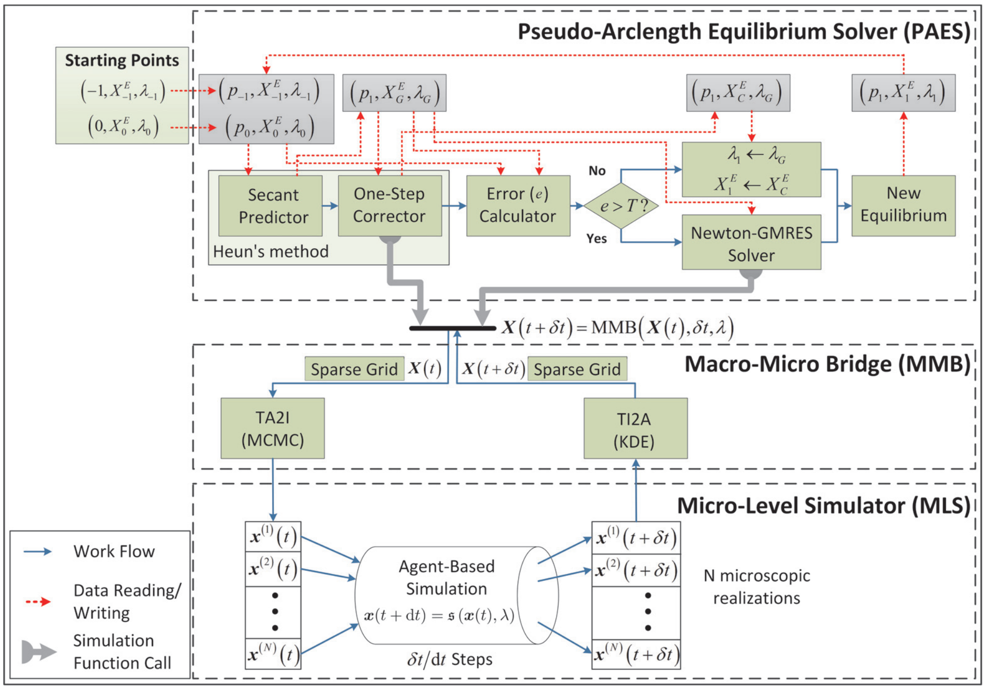
Pseudo-Arclength Equilibrium Solver (PAES)
Starting Points
$(-1, X^E_{-1}, \lambda_{-1})$
$(0, X^E_0, \lambda_0)$
$(p_{-1}, X^E_{-1}, \lambda_{-1})$
(p_0, X^E_0, λ_0)
(p_1, X^E_G, λ_G)
(p_1, X^E_C, λ_G)
(p_1, X^E_1, λ_1)
Secant Predictor
One-Step Corrector
Heun's method
Error (e) Calculator
$e > T$?
No
Yes
$\lambda_1 \leftarrow \lambda_G$
$X^E_1 \leftarrow X^E_C$
Newton-GMRES Solver
New Equilibrium
$X(t+\delta t) = \mathrm{MMB}(X(t), \delta t, \lambda)$
Macro-Micro Bridge (MMB)
Sparse Grid $X(t)$
$X(t+\delta t)$ Sparse Grid
TA2I (MCMC)
TI2A (KDE)
Micro-Level Simulator (MLS)
$x^{(1)}(t)$
$x^{(2)}(t)$
$x^{(N)}(t)$
Agent-Based Simulation
$x(t+\mathrm{d}t) = \mathfrak{s}(x(t), \lambda)$
$\delta t/\mathrm{d}t$ Steps
$x^{(1)}(t+\delta t)$
$x^{(2)}(t+\delta t)$
$x^{(N)}(t+\delta t)$
N microscopic realizations
Work Flow
Data Reading/ Writing
Simulation Function Call

Figure 1: Design of the Framework

where $Att^i \subseteq att^i$ and $S^i \subseteq s^i$ $(i = 1, \ldots, M)$ stand for the selected static attributes and dynamic states. $P^i(Att^i, S^i)$ denotes the probability distribution function for the ith agent type. $S_{Env} \subseteq s_{Env}$ represents the selected environment states. In addition, we write the complement of S^i and S_{Env} with respect to s^i and s_{Env} as $\bar{S}^i$ and $\bar{S}_{Env}$, respectively.

Our Framework

Fig. 1 illustrates the structure of the proposed framework which consists of three layers, namely, pseudo-arclength equilibrium solver (PAES), macro-micro bridge (MMB) and micro-level simulator (MLS). Our framework is designed for tracking the movement of the macroscopic equilibrium states $\boldsymbol{X}^E$ when the parameter λ continuously varies in the interval space $[\lambda_{min}, \lambda_{max}]$. It requires two starting points as inputs which are computed through Monte Carlo experiments. The values -1 and 0 in the diagram stand for the pseudo-arclength p which will be introduced in Section 4.2. Once the two equilibrium states at $p_{\text{-}1}$ and p_0 are given, Heun's method (Süli and Mayers 2003) is employed to compute the new equilibrium at p_1. However, since ABS is not differentiable, the secant direction generated by the two input points acts as an approximation of the derivative (Secant Predictor). Afterwards, the error e of the new equilibrium state is computed (Error Calculator). If the error is smaller than a threshold T, the output of Heun's method is accepted; otherwise, the iterative Newton-GMRES solver is utilized to reduce the error. The new equilibrium $\left(p_1, \boldsymbol{X}^E_1, \lambda_1\right)$ at p_1 is obtained and used as the new starting point for computing the equilibrium at p_2. This recursive process is computed until the entire parameter space is fully explored.

In PAES, both the one-step corrector and Newton-GMRES solver need to compute the macroscopic evolution dynamics of ABS. However, the macroscopic evolutionary dynamics of ABS usually cannot be explicitly denoted by mathematical equations. Thus, in our framework, we approximate the required macroscopic evolution through short bursts of microscopic simulation. This is achieved by developing the MMB module to bridge the micro- and macro-levels. PAES can easily obtain macroscopic evolution results by exploiting MMB. We focus on the case where the macroscopic variables are the distributions of agents' attributes. Hence, an algorithm based on the Markov Chain Monte Carlo (MCMC) method is used to generate N sets of microscopic realizations conforming to the macroscopic distributions in MMB (TA2I Module). Meanwhile, another algorithm based on the Kernel Density Estimation (KDE) technique is responsible for computing the macroscopic distributions based on microscopic simulation results (TI2A Module). In this way, the transformation between the micro- and macro-levels is achieved. In addition, for continuous macroscopic variables, the sparse grid acts as a better choice than the evenly distributed histogram for depicting the macroscopic distributions because it requires less number of points. The bottom layer of our framework is MLS which encapsulates the microscopic simulation. When N realizations are generated by MMB, MLS runs the microscopic simulation for $\delta t/\mathrm{d}t$ steps and feeds back the results to MMB. The detailed procedures of MMB and PAES will be described in the following subsections.

Algorithm 1: Macro-Micro Bridge

```
    input : $\boldsymbol{X}_0 = \left[P_0^1, \ldots, P_0^M, S_{Env,0}\right], \delta t, \lambda$
    output: $\boldsymbol{X}_{\delta t} = \left[P_{\delta t}^1, \ldots, P_{\delta t}^M, S_{Env,\delta t}\right]$
1   for $i \leftarrow 1$ to $M$ do
2       for $j \leftarrow 1$ to $n_i$ do
3           $S_P \leftarrow$ MCMC_Init($P_0^i(S^i, Att_j^i)$)
4           for $k \leftarrow 1$ to $N_b$ do
5               $S_P \leftarrow$ MCMC_Transition($P_0^i(S^i, Att_j^i), S_P$)
6           for $k \leftarrow 1$ to $N$ do
7               $S_P \leftarrow$ MCMC_Transition($P_0^i(S^i, Att_j^i), S_P$)
8               $S_j^{i(k)} \leftarrow S_P, \bar{S}_j^{i(k)} \leftarrow$ Initial Value
9               $s_j^{i(k)} \leftarrow \left(S_j^{i(k)}, \bar{S}_j^{i(k)}\right)$
10  for $k \leftarrow 1$ to $N$ do
11      $S_{Env}^{(k)} \leftarrow S_{Env,0}, \bar{S}_{Env}^{(k)} \leftarrow$ Initial Value
12      $\boldsymbol{x}^{(k)} \leftarrow \left[(s_j^i)_{j\in[1,n_i]}^{i\in[1,M]}, \left(S_{Env}^{(k)}, \bar{S}_{Env}^{(k)}\right)\right]$
13      for $i \leftarrow 1$ to $\delta t/\mathrm{d}t$ do
14          $\boldsymbol{x}^{(k)} \leftarrow s(\boldsymbol{x}^{(k)}, \lambda)$
15  for $i \leftarrow 1$ to $M$ do
16      Point_Set$^i \leftarrow \left[(\boldsymbol{x}^{(k)}.Att_j^i, \boldsymbol{x}^{(k)}.S_j^{i(k)})_{j\in[1,n_i]}^{k\in[1,N]}\right]$
17      for $l \leftarrow 1$ to $N_{SG}^i$ do
18          $P_l^i \leftarrow$ KDE(SG$_l^i$, Point_Set$^i$)
19      $P_{\delta t}^i \leftarrow [P_1^i, \ldots, P_{N_{SG}^i}^i]$
20  $S_{Env,\delta t} \leftarrow \frac{1}{N}\sum_{k=1}^N \boldsymbol{x}^{(k)}.S_{Env}$
21  $\boldsymbol{X}_{\delta t} = \left[P_{\delta t}^1, \ldots, P_{\delta t}^M, S_{Env,\delta t}\right]$
```

Macro-Micro Bridge

Due to the complexity of ABS, it is usually impossible to denote the macroscopic probability distributions $P(Att^i, S^i)$ with mathematical equations. For discrete states and attributes, it is straightforward to use histogram to represent a distribution. However, for continuous cases, the distribution function usually needs to be evenly segmented into a discrete grid that approximately represents $P(Att^i, S^i)$ with the histogram. This involves $O(N^{Dim})$ degrees of freedom, where Dim denotes the dimensionality of the distribution and N denotes the number of points in one dimension. Thus, the required number of points for approximating the distribution grows up exponentially as the dimensionality increases.

The differentiability of $P(Att^i, S^i)$ in our framework provides a better choice to achieve the approximation of continuous macroscopic probability distributions. This is ensured by the KDE-based transformation algorithm which will be introduced later in this section. Under the condition of differentiability, the sparse grid can be used to leverage the exponential freedom degrees $O(N^{Dim})$ of the evenly distributed grid, since it only requires $O(N \cdot (\log N)^{Dim-1})$ degrees of freedom. Sparse grid divides the space in a hierarchical way and approximates the desired distribution function with linear basis functions. Sparse Grid Interpolation Toolbox is used in the experiments to calculate the coordinates of the grid points and conduct interpolation. More details on sparse grid can be found in (Bungartz and Griebel 2004).

Algorithm 1 shows the basic procedures of MMB. It is designed to conduct the macroscopic evolution while keeping the details of the microscopic simulation at the micro-level. The inputs of the algorithm are: the initial values of macroscopic variables $\boldsymbol{X}_0$, the macroscopic step δt and the parameter value λ. The output is the macroscopic evolution result $\boldsymbol{X}_{\delta t}$, namely, the values of macroscopic variables after δt. MMB starts with the MCMC algorithm to generate N realizations conforming to $P_0^i(S^i, Att_j^i)$ for each agent (line 1-line 9). MCMC computes samples from the desired probability distribution through constructing a Markov chain whose equilibrium distribution is the same as the desired distribution. It randomly selects a point as the starting point for the state transition process of the Markov chain (line 3). As the initial point is not chosen based on the desired distribution, MCMC conducts the burn-in process through generating N_b steps of state transitions, where N_b is a manually defined large number (line 4-line 5). After the burn-in process, required microscopic states can be obtained through conducting N steps of state transitions (line 6-line 8). A detailed description of MCMC can be found in (Andrieu et al. 2003). In this framework, since the static attribute vector Att^i remains constant during the simulation run, the conditional probability distribution $P_0^i(S^i|Att_j^i)$ should be used for generating the microscopic realizations. Considering that MCMC only requires the probability ratio of two neighboring points, the probability distribution $P_0^i(S^i, Att_j^i)$ is used to replace the conditional distribution. Otherwise, $\int P_0^i(S^i, Att_j^i)\mathrm{d}S^i$ is required for computing the conditional distribution from the joint distribution. In the algorithm, the ignored microscopic state vector $\bar{S}^i$ needs to be set as manually selected initial values (line 8). The error in $\bar{S}^i$ can be reduced by the simulation as the slow-changing states gradually dominate and the macroscopic states move towards the emergent equilibrium (Kevrekidis and Samaey 2009).

After generating the required number of microscopic realizations, MMB calls MLS to run $\delta t/\mathrm{d}t$ steps of simulation for each realization (line 10-line 14). Then, MMB extracts the simulation results as a point set with $N \times n_i$ dimensions, followed by computing the probability value at each sparse grid point using the KDE algorithm (line 15-line 19):

$$\mathrm{KDE}\left(\mathrm{SG}_l^i, \mathrm{Point\_Set}^i\right) = \frac{1}{n_i N h}\sum_{k=1}^{N}\sum_{j=1}^{n_i} K\left(\frac{\mathrm{SG}_l^i - (\boldsymbol{x}^{(k)}.Att_j^i, \boldsymbol{x}^{(k)}.S_j^{i(k)})}{h}\right) \quad (5)$$

where $l = 1, 2, \ldots, N_{SG}^i$. N_{SG}^i stands for the required number of grid points for the ith type agent. SG_l^i represents the coordinates of the grid point. $K(\cdot)$ stands for the kernel function and h is the bandwidth. According to Equation 5, if the kernel function has bounded derivatives at any order, the obtained probability distribution should also have the same property. In this work, we have adopted RBF kernel which is differentiable at any order. Thus, the differentiability of $P(Att^i, S^i)$ can be ensured, which lays the foundation for using the sparse grid. As for the selection of h, there are many methods listed in (Jones, Marron, and Sheather 1996). In the two testbeds used by us, we use a fixed bandwidth. The last two steps of the algorithm compute the macroscopic

Algorithm 2: Pseudo-Arclength Equilibrium Solver

```
   input : $(p_{-1}, \boldsymbol{X}^E_{-1}, \lambda_{-1})$, $(p_0, \boldsymbol{X}^E_0, \lambda_0)$: the two starting points;
           $\delta p$: the step length of $p$;
           $[\lambda_{min}, \lambda_{max}]$: the parameter interval
   output: $(p_1, \boldsymbol{X}^E_1, \lambda_1)$,$(p_2, \boldsymbol{X}^E_2, \lambda_2) \dots$
1  for $i \leftarrow 1$ to $N_{\mathrm{PAES}}$ do
2      $p_i \leftarrow p_{i-1} + \delta p$
3      $\boldsymbol{X}^E_G \leftarrow \boldsymbol{X}_{i-1} + \delta p(\boldsymbol{X}_{i-1} - \boldsymbol{X}_{i-2})/(p_{i-1} - p_{i-2})$
4      $\boldsymbol{X}^E_G \leftarrow$ Distribution-Check($\boldsymbol{X}^E_G$)
5      $\lambda_G \leftarrow \lambda_{i-1} + \delta p(\lambda_{i-1} - \lambda_{i-2})/(p_{i-1} - p_{i-2})$
6      $\boldsymbol{X}_C \leftarrow$ MMB($\boldsymbol{X}^E_G, \delta t, \lambda_G$)
7      $\delta_X \leftarrow \boldsymbol{X}_C - \boldsymbol{X}_G$
8      $\delta_G \leftarrow$ PA_Error($\boldsymbol{X}^E_G, \boldsymbol{X}^E_{i-1}, \lambda_G, \lambda_{i-1}, p_i, p_{i-1}$);
9      $e \leftarrow \frac{1}{\sqrt{length(\delta_X)+1}} || [C \cdot \delta_X, \delta_G] ||_2$
10     if $e \leq T$ then
11         $(\boldsymbol{X}^E_i, \lambda_i) \leftarrow (\boldsymbol{X}_C, \lambda_G)$
       else
12         $(\boldsymbol{X}^E_i, \lambda_i) \leftarrow$ Newton-GMRES($p_i, \boldsymbol{X}^E_G, \lambda_G$)
13     if $\lambda_i \notin [\lambda_{min}, \lambda_{max}]$ then
           break
```

environment states $S_{Env,\delta t}$ by averaging all the N realizations and assemble the macroscopic variables (line 20-line 21). Since MCMC and KDE can easily handle discrete variables using the given expression, we consider the more challenging case with continuous variables in the testbeds.

Pseudo-Arclength Equilibrium Solver

According to the condition of emergent equilibrium (Equation 3), the equilibrium states $\boldsymbol{X}^E$ to be solved by our framework need to fulfill the condition shown in Equation 6.

$$\delta_X := \boldsymbol{X}^E - \mathrm{MMB}(\boldsymbol{X}^E, \delta t, \lambda) = 0 \qquad (6)$$

According to the NC theory, relying only on Equation 6 may lead the numerical computation to get stuck at singular points in the parameter space (Shroff and Keller 1993). Thus, the pseudo-arclength p which represents the arclength of equilibrium movement in the joint multi-dimensional space of states and the parameter is introduced with the continuity constraint $\delta_G(p, \boldsymbol{X}, \lambda) = 0$ (Shroff and Keller 1993). Therefore, the equilibrium should fulfill:

$$e := \frac{1}{\sqrt{length(\delta_X) + 1}} || [C \cdot \delta_X, \delta_G] ||_2 = 0 \qquad (7)$$

where C is the parameter used to balance the scale of δ_X and δ_G. In the implementation, we only require $e \leq T$, where T is the threshold chosen based on the maximum variation of equilibrium states at the two starting points.

Algorithm 2 shows the procedures of PAES. It is designed for adaptively exploring the desired parameter interval based on the pseudo-arclength method. The inputs to the algorithm include the step length of p, the parameter interval $[\lambda_{min}, \lambda_{max}]$, and two starting points obtained from using MCE at the selected parameter values. The output of PAES is a set of points that represent the emergent equilibrium for different λ values. Since the algorithm adaptively adjusts the parameter λ, the exact number of output points cannot be decided before running the program.

PAES begins with computing the pseudo-arclength value p_i for the new equilibrium using the starting points p_{i-1} and p_{i-2} (line 2). Then, the secant method is employed to predict the states and parameter's value of the new equilibrium (line 3 and line 5). Since the secant prediction may produce a negative probability value, the Distribution-Check function is utilized to set the negative values to be 0 (line 4). Based on the predicted equilibrium states, a simple one-step correction is conducted by calling MMB to solve the macroscopic evolution dynamics (line 6). The error of the newly obtained equilibrium is computed using Equation 7 (line 7-line 9). For the secant prediction, δ_G can be computed using the following equation proposed by (Shroff and Keller 1993):

$$\delta_G = \frac{1}{s_i - s_{i-1}} \left[||\boldsymbol{X}^E_G - \boldsymbol{X}_{i-1}||^2_2 + (\lambda_G - \lambda_{i-1})^2 - (s_i - s_{i-1})^2 \right] \qquad (8)$$

If the error e is smaller than the threshold T, the corrected states $\boldsymbol{X}_C$ and the secant prediction of parameter λ_G will be accepted as the new equilibrium (line 11). Otherwise, the Newton-GMRES solver (Kelley 2003) is used to solve the equations $[C \cdot \delta_X, \delta_G]^T = 0$ (line 12). This solver avoids the computation of Jacobian matrix by using the matrix-free iterative eigensolver to construct the gradient direction. Through the Newton iterations, the error will be reduced to be smaller than T, and the solution of the target equations which is also the new equilibrium is obtained. Then, the newly obtained equilibrium will be used to compute next equilibrium, and this recursive process terminates when λ_i goes out of the desired parameter space (line 13) or the maximum rounds of computation N_{PAES} are reached (line 1).

Experiments

In order to evaluate the performance of our framework, we have applied it to two testbeds. The first one is the opinion dynamics model, and the second one is the labour market model with more complex interaction rules from the domain of economics (Fagiolo, Dosi, and Gabriele 2004).

Opinion Dynamics Model

Opinion dynamics model considers a group of agents who interact with each other to reach an agreement. Each agent has an opinion $x \in [0, 1]$. Agents also have a certain level of confidence on the the accuracy of their opinions, which is depicted by the confidence bound ϵ. When interacting with each other, agents only allow their opinions to be swayed by those who hold opinions within the confidence bound of theirs. In other words, if the opinion difference of two agents is smaller than the confidence bound, they can reach a consensus after interaction (for example, face-to-face discussion); otherwise, their opinions remain unchanged. Formally, the interaction among agents is defined as a random process (Equation 9) in which two agents i and j are randomly selected to perform the action at each time

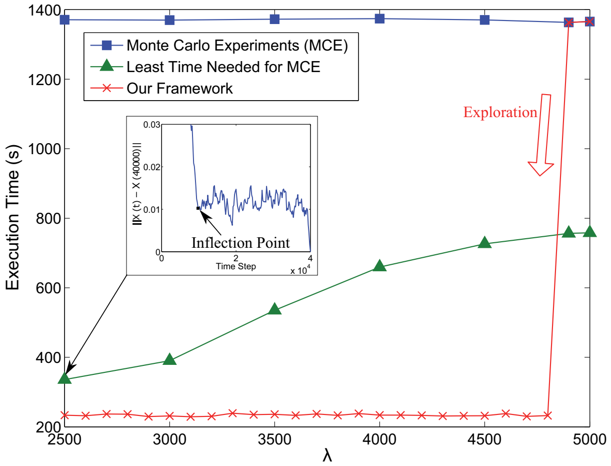

Figure 2: Execution Time Comparison

step (Hegselmann and Krause 2002).

$$x^k(t+1) = \begin{cases} \frac{x^j(t)+x^i(t)}{2} & ||x^i(t) - x^j(t)|| \le \epsilon \\ x^k(t) & otherwise \end{cases} \tag{9}$$

where k can be i or j.

We add an additional attribute, age (denoted as l), to agents to make the testbed more practical, and the state of agent i can be described as $< l^i, x^i >$. In our model, old agents may die and leave the system at any time step while new-born ones join the system to keep the total number of agents constant. We assume the probability of an agent living to an age of l to follow Weibull distribution, $P(l;\lambda,k) = \frac{k}{\lambda}\left(\frac{l}{\lambda}\right)^{k-1} e^{-(l/\lambda)^k}$, where $k > 0$ is the shape parameter and $\lambda > 0$ is the scale parameter (Rinne 2008). In this model, these two parameters reflect the living mode and the average age of the population, respectively. Thus, the probability for a agent to leave during l and $l+1$ is

$$P_L([l,l+1];\lambda,k) = 1 - \frac{1 - \int_{a=0}^{a=l+1} P(a;\lambda,k)}{1 - \int_{a=0}^{a=l} P(a;\lambda,k)} \tag{10}$$

By setting k to be 1, we adopt the constant hazard model without “burn-in” nor “wear” process. In addition, we set 5λ as the upper bound of an agent’s age. If an agent’s age reaches the upper bound, it will be forced to die.

In this experiment, the joint distribution $P(l,x)$ is selected to be the macroscopic variable. We study the effects of changing the scale factor λ. The total number of agents is set to 5000. The macroscopic step δt is chosen as 6000 microscopic simulation steps, and the number of microscopic realizations is set to 10. The confidence bound ϵ is chosen as 0.5. In executing our experiment, we use MATLAB 2013a running on Intel Xeon CPU E5-1650, 16 GiB RAM and Ubuntu 14.04LTS operating system. Fig. 2 shows the computation time of running this testbed through our framework against that through MCE. The two starting points for our framework are selected as $\lambda = 5000$ and $\lambda = 4900$ which correspond to the two rightmost square points in the figure. They are computed using MCE in our framework. Our framework starts to automatically explore the desired parameter space $[2500, 5000]$ by conducting prediction and correction at the third points. Since the equilibrium moves smoothly as the parameter changes in opinion dynamics, Heun’s method was selected by our framework at all the auto-selected parameter values, which leads to the constant parameter value interval and the similar execution time for all values.

Table 1: Results for B-test (Significance Level $\alpha = 0.1$)

λ	2500	3000	3500	4000	4500
p-value	0.6738	0.3980	0.5316	0.3875	0.2254

We manually conduct MCE for comparison at some parameter values and use B-test (Zaremba, Gretton, and Blaschko 2013) to examine whether the equilibrium states computed by MCE and our framework are the same. The obtained equilibrium states are tested for the following null hypothesis $\mathcal{H}_0$ and alternative hypothesis $\mathcal{H}_A$:

$$\mathcal{H}_0 : PDF_O = PDF_{MCE}, \mathcal{H}_A : PDF_O \neq PDF_{MCE}$$

where PDF_O and PDF_{MCE} denote the probability distribution function at the equilibriums computed by our framework and MCE, respectively. The significance level α is chosen as 0.1. Thus, if the computed p-value in B-test is bigger than 0.1, the null hypothesis will be accepted; otherwise the alternative hypothesis will be accepted. The total number of simulation steps is manually set to be a very large number (40,000 in this case) to ensure the convergence of the system. From the B-test results shown in Table 1, we can conclude that there is no significant difference between the equilibrium states solved by two methods. Meanwhile, from the comparison of computation time in Fig. 2, we can conclude that our framework can significantly reduce the time needed for solving the equilibrium states. Furthermore, we analyze the error variation during MCE in details, and take the inflection point as the mark for reaching the convergence. The execution time required for reaching the inflection point stands for the smallest time needed for the convergence of MCE. As shown in the figure, the smallest time needed by MCE is much longer than the execution time of our framework at every selected point.

Labour Market Model

The labour market model describes an economy with F firms and N workers. A firm i in the model is denoted by $< q_{it}, w_{it}, w^s_{it} >$, where q_{it}, w_{it} and w^s_{it} represent the production, the real wage offered and the “satisficing” wage, respectively. A worker j is described by $< w^s_{jt}, w^R_j >$, where w^s_{jt} and w^R_j stand for the “satisficing” and reservation wage, respectively. The continuous interaction between these two types of agents lead the system to reach a macroscopic equilibrium state. Specifically, at time step t, the output of firm i is assumed to be linearly proportional to the labour used:

$$q_{it} = \alpha_{it} n_{it} \tag{11}$$

where α_{it} is the labour productivity and n_{it} is the number of workers hired. Thus, the profit can be computed as

$$\pi_{it} = (p_t \alpha_{it-1} - w_{it}) n_{it} \tag{12}$$

where w_{it} is the contractual wage offered by firm i. Any firm i has a "satisficing" wage w^s_{it} at time t which the firm wants to offer to its workers. Any worker j also has a 'satisficing" wage w^s_{jt} which he wants to get from firms. Furthermore, worker j can only accept contractual wages if the contractual wages are greater or equal to their reservation wage w^R_j, which we assume to be a constant for simplicity. The interaction between workers and firms at time step t can be summarized as the following six steps and the more detailed explanation of the interaction process can be found in (Fagiolo, Dosi, and Gabriele 2004):

1. Firm i decides the number of vacancies v_{it} as $\lceil \frac{p_{t-1}q_{it-1}}{w_{it-1}} \rceil$, where $\lceil \cdot \rceil$ stands for the "ceiling" operation.
2. Each worker j goes to the market and visits firm i with a probability proportional to w_{it-1}. If the selected firm has places available in the queue, the worker gets in and demands a wage equal to w^s_{jt-1}.
3. At time step t, firm i observes m_{it} workers in the queue. Thus, it adjusts its satisficing wage according to the expected wage of workers:
$$w_{it} = \beta w^s_{it-1} + (1-\beta)\frac{1}{m_{it}}\sum_{h=1}^{m_{it}} m_{it} w^s_{j_h t-1} \tag{13}$$
If $w_{it} \geq w^R_j$, worker j in the queue will accept the job, and set w^s_{jt-1} to be w_{it}. Similarly, a firm which has filled at least a job opening will replace w^s_{it-1} with w_{it}.
4. After hiring workers, firms produce the products according to Equation 11. Then, the products are sold with the "pseudo-Walrasian" price p_t which is given by $p_t = W_t/Q_t$. Here, $Q_t = \sum_{i=1}^{F} q_{it}$ is the aggregate output and $W_t = \sum_{j=1}^{N} w_{jt}$ is the total wage.
5. According to Equation 12, the profit of every firm can be computed. Then, firms with negative profits will be weeded out and new entrants with the average "characteristics" are added into the market.
6. Assume that each firm has an invariant desired ratio of filled to opened jobs $\rho_i \in (0, 1]$. The current filling ratio $r_{it} = n_{it}/v_{it}$ is compared with the desired ratio to adjust the satisficing wage as:
$$w^s_{it} = \begin{cases} w^s_{it-1}(1+|Y|), \text{if } r_{it} < \rho_i \\ w^s_{it-1}(1-|Y|), \text{if } r_{it} \geq \rho_i \end{cases} \tag{14}$$
where Y follows the standard normal distribution. Meanwhile, the satisficing wage w^s_{it} for worker j is updated as:
$$w^s_{jt} = \begin{cases} w^s_{jt-1}(1-|Y|), \text{unemployed} \\ w^s_{jt-1}(1+|Y|), \text{employed} \end{cases} \tag{15}$$
Besides, if w^s_{jt} becomes less than w^R_j, it will be directly set as the latter.

At every time step, all agents repeatedly conduct interaction according to these 6 steps, and finally, the labour market will converge to a certain macroscopic equilibrium.

In this model, we also use MCE and our framework to compute the equilibrium states for comparison. The total number of firms and works is set to be 200 and 1000, respectively. The macroscopic time step δt is selected as 50 microscopic simulation steps, and the number of microscopic realizations is set to 50. For simplicity, we only use the distribution of w^s to describe firms at the macro-level because q and w are computed without considering the previous values at every step. Thus, q and w are reinitialized to be the initial values for every step of macroscopic evolution. Workers' macroscopic states are represented by the joint distribution $P(w^s, w^R)$. $\rho = 0.80$ and $\rho = 0.81$ are used as the two starting points. δs is set to be 0.1. In order to avoid the effects of macroscopic variable simplification in comparison, we also reinitialize q and w every 50 simulation steps in MCE.

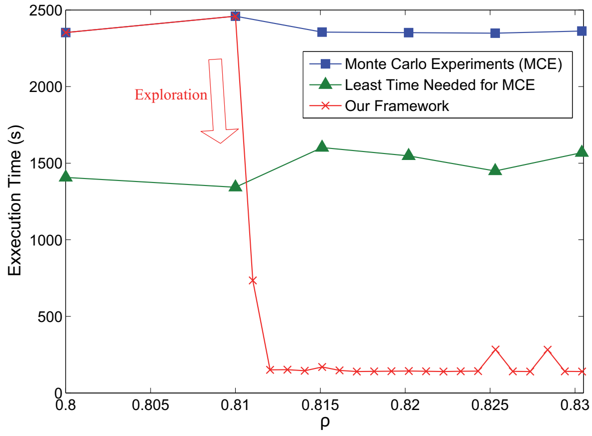

Figure 3: Execution Time Comparison

Table 2: Results for B-test (Significance Level $\alpha = 0.1$)

ρ	0.8151	0.8202	0.3080	0.4988
p-value for Firms	0.7974	0.4542	0.5316	0.3875
p-value for Workers	0.4683	0.5593	0.4969	0.5144

Table 2 shows the B-test results for the macroscopic distributions of firms and workers, which illustrates that there is no significant difference between the equilibrium results of our framework and the MCE. Meanwhile, Fig. 3 shows the execution time comparison between the two methods. As shown in the figure, our framework performs much better than the traditional MCE (the total number of steps is set to be 1000). Compared with the opinion dynamics model, the labour market model is a more complex, leading the framework to use the Newton-GMRES solver to reduce the error at some points. This causes an increase in the execution time at the corresponding points.

Conclusion

In this paper, we propose a novel three-layer framework to help modelers compute the emergent equilibrium over the parameter space more efficiently. Using two previously obtained equilibrium states as the starting points, PAES in our framework can adaptively explore the parameter space at the macro-level, while the refined ABS simulation is executed at the micro-level. To bridge the micro- and macro-levels, MMB is introduced between PAES and the microscopic simulation. It approximates the macroscopic evolution dynamics through short bursts of a large amount of microscopic simulation. Experimental results on two models confirm that our framework can achieve a much higher efficiency in computing the emergent equilibrium of ABS comparing with the traditional MCE. For future work, instead of relying on the probability distribution, we will apply data mining techniques, such as Diffusion Maps, to generalize our framework. We will also improve the micro-level solver for more complex macroscopic variables and implement a proactive algorithm to detect sharp equilibrium state changes which can cause the failure of the NC theory.

Acknowledgments

This work was conducted within the Rolls-Royce@NTU Corporate Lab with support from the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme. The authors thank Prof. Sinno Jialin Pan, Dr. Donghun Kang and PhD student Yukun Ma for valuable discussions.

References

Allan, R. J. 2010. *Survey of agent based modelling and simulation tools*. Swindon, United Kingdom: Science & Technology Facilities Council.

Allgower, E. L., and Georg, K. 2003. *Introduction to numerical continuation methods*, volume 45. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Andrieu, C.; De Freitas, N.; Doucet, A.; and Jordan, M. I. 2003. An introduction to mcmc for machine learning. *Machine learning* 50(1-2):5–43.

Bert, F. E.; Rovere, S. L.; Macal, C. M.; North, M. J.; and Podestá, G. P. 2014. Lessons from a comprehensive validation of an agent based-model: The experience of the pampas model of argentinean agricultural systems. *Ecological Modelling* 273:284–298.

Bungartz, H.-J., and Griebel, M. 2004. Sparse grids. *Acta numerica* 13:147–269.

Chan, T. F., and Keller, H. 1982. Arc-length continuation and multigrid techniques for nonlinear elliptic eigenvalue problems. *SIAM Journal on Scientific and Statistical Computing* 3(2):173–194.

Fagiolo, G.; Dosi, G.; and Gabriele, R. 2004. Matching, bargaining, and wage setting in an evolutionary model of labor market and output dynamics. *Advances in Complex Systems* 7(02):157–186.

Hegselmann, R., and Krause, U. 2002. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of Artificial Societies and Social Simulation* 5(3).

Jones, M. C.; Marron, J. S.; and Sheather, S. J. 1996. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association* 91(433):401–407.

Kelley, C. T. 2003. *Solving nonlinear equations with Newton's method*, volume 1. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Kevrekidis, I. G., and Samaey, G. 2009. Equation-free multiscale computation: Algorithms and applications. *Annual review of physical chemistry* 60:321–344.

Liu, P. 2013. *Equation-Free Analysis for Agent-Based Computation*. Ph.D. Dissertation, Princeton University.

North, M. J., and Macal, C. M. 2007. *Managing business complexity: discovering strategic solutions with agent-based modeling and simulation*. Oxford, United Kingdom: Oxford University Press.

Olaru, D., and Purchase, S. 2014. Rethinking validation: Efficient search of the space of parameters for an agent-based model. *Australasian Marketing Journal (AMJ)* 22(1):60–68.

Picault, S., and Mathieu, P. 2011. An interaction-oriented model for multi-scale simulation. In *Proceedings of the International Joint Conference on Artifical Intelligence (IJCAI)*, 332–337. Barcelona, Spain: AAAI Press.

Rinne, H. 2008. *The Weibull distribution: a handbook*. Boca Raton, Florida.: CRC Press.

Shroff, G. M., and Keller, H. B. 1993. Stabilization of unstable procedures: the recursive projection method. *SIAM Journal on numerical analysis* 30(4):1099–1120.

Siettos, C. 2014. Coarse-grained computational stability analysis and acceleration of the collective dynamics of a monte carlo simulation of bacterial locomotion. *Applied Mathematics and Computation* 232:836–847.

Stonedahl, F., and Wilensky, U. 2010. Finding forms of flocking: Evolutionary search in abm parameter-spaces. In *Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS) Workshop on Multi-Agent-Based Simulation (MABS)*. Toronto, Canada: Springer. 61–75.

Süli, E., and Mayers, D. F. 2003. *An introduction to numerical analysis*. Cambridge, United Kingdom: Cambridge University Press.

Teose, M.; Ahmadizadeh, K.; O'Mahony, E.; Smith, R. L.; Lu, Z.; Ellner, S. P.; Gomes, C.; and Grohn, Y. 2011. Embedding system dynamics in agent based models for complex adaptive systems. In *Proceedings of the International Joint Conference on Artifical Intelligence (IJCAI)*, 2531–2538. Barcelona, Spain: AAAI Press.

Tsoumanis, A. C.; Siettos, C. I.; Bafas, G. V.; and Kevrekidis, I. G. 2010. Equation-free multiscale computations in social networks: from agent-based modeling to coarse-grained stability and bifurcation analysis. *International Journal of Bifurcation and Chaos* 20(11):3673–3688.

Zaremba, W.; Gretton, A.; and Blaschko, M. B. 2013. B-test: A non-parametric, low variance kernel two-sample test. *CoRR* abs/1307.1954.