INSTITUTE
OF ECONOMICS

Scuola Superiore
Sant'Anna

LEM | Laboratory of Economics and Management

Institute of Economics
Scuola Superiore Sant'Anna

Piazza Martiri della Libertà, 33 - 56127 Pisa, Italy
ph. +39 050 88.33.43
institute.economics@sssup.it

LEM
WORKING PAPER SERIES

Automated and Distributed Statistical Analysis of Economic Agent-Based Models

Andrea Vandin [a]
Daniele Giachini [a]
Francesco Lamperti [a,c]
Francesca Chiaromonte [a,b]

[a] Institute of Economics and EmbeDS Department, Scuola Superiore Sant'Anna, Pisa, Italy.
[b] Department of Statistics and Huck Institutes of the Life Sciences, Penn State University, USA.
[c] RFF-CMCC European Institute on Economics and the Environment, Milan, Italy.

2020/31 November 2020
ISSN(ONLINE) 2284-0400

Automated and Distributed Statistical Analysis of Economic Agent-Based Models

Andrea Vandin[a], Daniele Giachini[a], Francesco Lamperti[a,c], Francesca Chiaromonte[a,b]

[a] *Institute of Economics and EMbeDS, Sant'Anna School of Advanced Studies, Pisa, Italy.*
[b] *Dept. of Statistics and Huck Institutes of the Life Sciences, Penn State University, USA*
[c] *RFF-CMCC European Institute on Economics and the Environment, Milan, Italy.*

Abstract

We propose a novel approach to the statistical analysis of simulation models and, especially, agent-based models (ABMs). Our main goal is to provide a fully automated and model-independent tool-kit to inspect simulations and perform counter-factual analysis. Our approach: (i) is easy-to-use by the modeller, (ii) improves reproducibility of results, (iii) optimizes running time given the modeller's machine, (iv) automatically chooses the number of required simulations and simulation steps to reach user-specified statistical confidence, and (v) automatically performs a variety of statistical tests. In particular, our framework is designed to distinguish the transient dynamics of the model from its steady state behaviour (if any), estimate properties of the model in both "phases", and provide indications on the ergodic (or non-ergodic) nature of the simulated processes – which, in turns allows one to gauge the reliability of a steady state analysis. Estimates are equipped with statistical guarantees, allowing for robust comparisons across computational experiments. To demonstrate the effectiveness of our approach, we apply it to two models from the literature: a large scale macro-financial ABM and a small scale prediction market model. Compared to prior analyses of these models, we obtain new insights and we are able to identify and fix some erroneous conclusions.

Keywords: ABM, Statistical Model Checking, Ergodicity analysis, Steady state analysis, Transient analysis, Warmup estimation, T-test and power, Prediction markets, Macro ABM

JEL Classification: C15, C18, C63, D53, E30

1. Introduction

In this article we present a model-independent and fully automated approach to the statistical analysis of simulation models and, especially, agent-based models (ABMs). Leveraging a tool-box of efficient algorithms to inspect simulations and perform model-based counter-factual analysis, our approach (i) is easy-to-use by the modeller, (ii) improves reproducibility of the results, (iii) distributes simulations across the cores of a machine or across computer networks, (iv) automatically chooses a sufficient number of simulations and simulation steps to reach a user-specified statistical confidence, and (v) automatically runs a variety of statistical tests that are often overlooked by practitioners. In particular, the proposed approach allows one to distinguish the *transient dynamics* of the model from its *steady state behaviour* (if any), to estimate properties of the model in both "phases", to check whether the ergodicity assumption is

Email addresses: `a.vandin@santannapisa.it` (Andrea Vandin), `d.giachini@santannapisa.it` (Daniele Giachini), `f.lamperti@santannapisa.it` (Francesco Lamperti), `f.chiaromonte@santannapisa.it` (Francesca Chiaromonte)

reasonable, and to equip the results with statistical guarantees, allowing for robust comparison of model behaviours' across computational experiments.

In the last two decades, the use of Agent-Based Models (ABMs) has spread across several fields – including ecology (Grimm and Railsback, 2013), health care (Effken et al., 2012), sociology (Macy and Willer, 2002), geography (Brown et al., 2005), medicine (An and Wilensky, 2009), research in bioterrorism (Carley et al., 2006), and military tactics (Ilachinski, 1997). In economics, ABMs contributed to the understanding of a variety of micro and macro phenomena (Tesfatsion and Judd, 2006). They provided an alternative environment for policy-testing in the aftermath of the last financial crisis, when more traditional approaches (e.g., dynamic stochastic general equilibrium models and computable general equilibrium models) failed (Fagiolo and Roventini, 2012, 2017). Moreover, they were recently used for macroeconomic forecasting with promising results (Delli Gatti and Grazzini, 2020).

The key advantage of ABMs is the flexibility they allow in modelling realistic micro-level behaviours (e.g., bounded rationality, routines, stochastic decision processes) and agents' interactions (e.g., imitation, network effects, spatial influence), which give rise to aggregate dynamics that are qualitatively different from those at the individual level (emergent properties). This advantage comes at the cost of model complexity, which typically prevents analytical treatment and forces the modeller to rely on numerical simulations.

Typically, little attention has been devoted to simulation protocols. Yet decisions about (i) how many steps to run, (ii) how many steps to "cut" up front as transient (aka, the warm-up period), and (iii) how many runs to perform under each parameter configuration, deeply influence an analysis and the reliability of its results. For example, statements like "*the results have been averaged over n simulations*" or "*we run a Monte Carlo exercise of size n*", without a proper justification for the choice of n, are rather ubiquitous (see e.g. Beygelzimer et al., 2012; Kets et al., 2014; Caiani et al., 2016; Lamperti et al., 2018, 2019; Dosi et al., 2019; Fagiolo et al., 2020). This can lead to ineffective estimates of model behaviour, with low precision and poor statistical confidence. While irrelevant for "thought experiments", these aspects deserve more attention when different policies are compared in counter-factual simulation experiments, or multiple parameter configurations are explored to discriminate among emerging behaviours. Secchi and Seri (2017) conducted a study on 55 ABMs published between 2010 and 2013 in high-quality management and organizational science journals. Their study showed that - in most cases - simulation exercises did not offer acceptable statistical quality[1], casting doubt on the results and their implications. The main cause of low statistical accuracy turned out to be an insufficient number (n) of simulations performed. Similarly, a poor handling of transient behaviours can distort results. As we will show in Section 7, discarding "*the initial w periods from each simulation to focus on the stationary behaviour*" (see, e.g., Kets et al., 2014) without a proper justification for the choice of w can lead to misleading conclusions about steady state properties. Furthermore, as we illustrate in Section 8, ergodicity tests are necessary in order to establish whether performing a steady state analysis makes sense at all.

In our opinion, these problems are due to the fact that the simulation-based analysis of ABMs (i.e., the inspection of models' simulations) is often *handcrafted*, resulting in a time-consuming and error-prone process (see also Lee et al., 2015). The simulation, operations research, and computer science communities have substantially advanced the engineering of such tasks, developing automatic techniques equipped with statistical guarantees (see, e.g., Law and Kelton, 2015,). While cross-disciplines fertilisation has recently increased (Dahlke et al., 2020), these developments are often overlooked by the so-called ACE (agent-based computational economics) community. In Section 2, we use the model by Grazzini (2012) to illustrate an example concerning the identification of transient dynamics.

In this article we introduce a novel fully-automated and engineered approach to ABMs inspection. We borrow

[1]The authors analyzed the power of t-tests in simulations on different model parametrizations.

the statistical model checking approach from computer science (Agha and Palmskog, 2018; Legay et al., 2019) to efficiently analyse stochastic simulation models and equip results with statistical guarantees. In particular, we propose novel, streamlined, parallelized and automated algorithms to carry out both *transient analysis* (estimating the average dynamics of the model at specific time points and characterizing the associated uncertainty) and *steady state analysis* (estimating the average dynamics of the model *on the long run* by estimating and removing the transient period) – and show that such algorithms are computationally efficient. We also equip our steady state analysis with a methodology for *ergodicity diagnostics*, which provides indications on whether the model behaves ergodically, and thus on the the reliability of the steady state analysis itself.

Our work contributes to two strands of the ACE literature. First, it complements many recent proposals for the validation of simulated models (see the surveys in Fagiolo et al., 2019; Lux and Zwinkels, 2018). For example, the method proposed by Guerini and Moneta (2017) for macro ABMs requires the model to be in a steady state as well as the removal of all observations belonging to the transient period, and calibration approaches based on simulated moments (Winker et al., 2007; Franke and Westerhoff, 2012; Grazzini and Richiardi, 2015), as well as recent Bayesian techniques (e.g. Grazzini et al., 2017), typically apply to ergodic models. From a different perspective, our transient analysis can evaluate multiple features at each time step – including the probabilities of observing certain patterns – and thus can support the use of validation metrics recently proposed in the literature (e.g. Barde, 2016; Lamperti, 2018a,b). Second, we contribute to the analysis of the complexities of ABMs' output (Lee et al., 2015; Mandes and Winker, 2017; Kukacka and Kristoufek, 2020) by providing fast and practical tools to inspect models with statistical guarantees (Secchi and Seri, 2017), and by complementing the proposals in Seri and Secchi (2017) for determining the adequate number of simulation runs to use. Finally, we offer an automated environment to carry out tests across experiments that are typical in the macro ABM literature (see e.g. Dosi et al., 2015).

We validate our approach on two models from the literature. [2] In Section 6 we replicate and enrich the transient analysis from Caiani et al. (2016) on a large scale benchmark stock flow consistent macro ABM. We optimize the number of simulations to reach a given (user-defined) level of statistical precision for each time point of interest. We show how this is necessary to establish, in a statistically sound manner, differences across model configurations – thereby facilitating counter-factual policy analysis. In Section 7 we perform a steady state analysis of the prediction market model of Kets et al. (2014). This model has been chosen because of its analytical tractability, which provides an effective ground truth against which we test our framework [3]. We show that an erroneous identification of the transient period led to misleading qualitative and quantitative results in the original simulation-based analysis by Kets et al. (2014); the number of long-run surviving agents and the relationships among market price and other model parameters were incorrectly characterized, and the agents' relative wealths were miscomputed. Instead, our framework allows us to correctly detect the transient period and correctly characterize the (analytically known) steady state properties by Bottazzi and Giachini (2019b). In Section 8 we also apply our methodology for ergodicity analysis to (non-ergodic) variants of this prediction market model, showing how it can be used to further increase the reliability of a steady state analysis. Finally, we highlight how the distributed nature of our algorithms allows us to automatically parallelise simulations in the cores of a computer without modifying the original (purely sequential) model. This affords, e.g., a 20x speed-up on a machine with 20 physical cores – in the case of the macro ABM model, this resulted in a decrease of analysis run-time from 15 days to about 16 hours.

From a technical perspective, our framework builds on MultiVeStA (Sebastio and Vandin, 2013; Gilmore et al.,

[2]Material for replicating the experiments presented in this paper is available at `https://github.com/andrea-vandin/MultiVeStA/wiki`

[3]The model has been analytically studied in Bottazzi and Giachini (2019b), proving asymptotic results about agents' wealth and market price.

2017), a statistical model checker that can be integrated with existing simulators to perform automated and distributed statistical analysis. MultiVeStA has been successfully applied in a wide range of domains including, e.g., highly-configurable systems (ter Beek et al., 2020, 2015), public transportation systems (Gilmore et al., 2014; Ciancia et al., 2016; Gilmore et al., 2017), biological systems (Gilmore et al., 2017), robotic scenarios with planning capabilities (Belzner et al., 2016, 2014), and crowd steering scenarios (Pianini et al., 2014). However, it has never been applied to the ABM domain. Further, its main focus so far was on transient analysis (without support for counterfactual analysis). Here we extend it to fully support the above mentioned tasks. [4]

The reminder of this article is organized as follows. Section 2 discusses the analysis of simulation output. Section 3 and 4 present our algorithms and methodology, respectively, and Section 5 introduces MultiVeStA. Sections 6 and 7 illustrate our transient and steady state analysis techniques using two ABM models from the literature. Section 8 showcases our methodology for ergodicity analysis. Section 9 demonstrates the run-time gains afforded by MultiVeStA's parallelization capabilities, and Section 10 provides some conclusions.

2. Analysis of simulation output

ABM analysis typically employs stochastic simulations, relying on Monte Carlo methods, to derive reliable estimates of the true model characteristics (Richiardi et al., 2006; Lee et al., 2015; Fagiolo et al., 2019).

Without loss of generality, one can represent an ABM as a mapping $map : I \to O$ from a set of input parameters I into an output set O. I is usually a multidimensional space spanned by the support of each parameter. O is typically larger and more complex, as it comprises time-series realizations of a very large number of micro- and macro-level variables. In most cases we can think of the output of an ABM as a discrete-time stochastic process $(\mathbf{Y}_t)_{t>0}$ describing the longitudinal evolution of a vector of variables of interest (e.g., the wealth of an agent, the GDP of a country, etc.). For simplicity, here we focus on the case in which $(\mathbf{Y}_t)_{t>0}$ contains only one time series of interest $(Y_t)_{t>0}$. However, our framework straightforwadly covers the concurrent analysis of multiple time series.

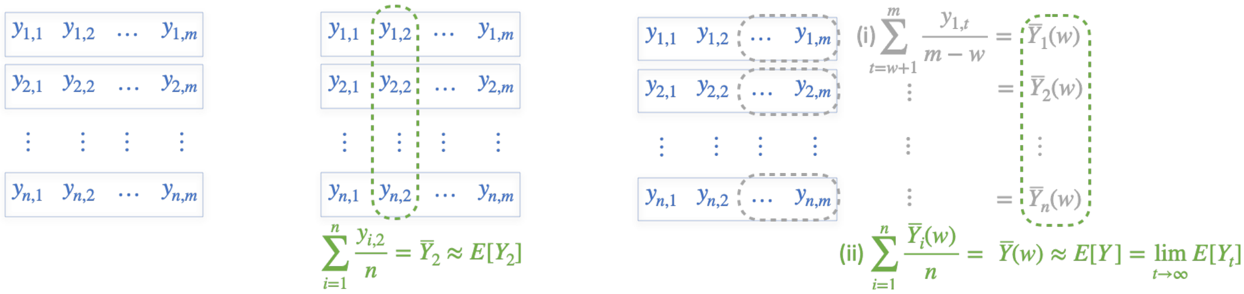
Figure 1(a) depicts n independent simulations of Y_t (one per row) each comprising $t = 1, \dots, m$ steps (one per column) [5]. The outcome of a simulation i is therefore a sequence $\{y_{i,1}, \dots, y_{i,m}\}$ denoting a realization of length m. Clearly, the observations within the same row i are not independent, while those in the same column t are independent and identically distributed (IID). Here we focus on two typical classes of properties:

- *Transient properties* concerning $E[Y_t]$; what is the expected value of a model's property at a given time t (or within a time range, or at the occurrence of a specific event)?
- *Steady state properties* concerning $E[Y] = \lim_{t\to\infty} E[Y_t]$; what is the expected value of a model's property at steady state (i.e. when the system has reached a statistical equilibrium, or after the initial warm-up period)?

An example of transient property is given in Section 6: *what is the expected unemployment rate in each of the first 400 quarters of a macro ABM?* In this example a transient analysis is particularly important because the model has been designed to study fluctuations in the quarters following a given initial condition. In contrast, an example of steady state property is given in Sections 7 and 8: *given a market with repeated sessions, what is the expected wealth of*

[4] MultiVeStA supports Java, R, C++, and Python simulators. The extension regards support for statistical tests (and their power) to compare different model parametrizations, and ex-novo development of steady state analysis. Furthermore, by applying it to two known ABM models, we also contribute to increasing the accessibility of ABMs and to the replicability of their results. MultiVeStA is maintained by one of the authors.

[5] By *independent simulations* we mean runs obtained from different random seeds that have been used for each replication, with the simulator status reset to an initial configuration at the beginning of each replication.



(a) Simulations (b) Transient analysis (c) Steady state analysis by *Replication and Deletion* (RD)

Figure 1: Transient and steady state analysis using n simulations of m steps each

each agent at steady state? In this example a steady state analysis is particularly important because the model has been designed to study problems of market selection and informative efficiency. Obviously a steady state analysis is meaningful only "around" a statistical equilibrium. This requires that $\lim_{t\to\infty} E[Y_t]$ exists and is finite. We first present two complementary techniques for steady state analysis that rely on such assumption, and then (in Section 4) combine them into a methodology for ergodicity diagnostics; that is, for assessing whether the assumption is reasonable or clearly violated [6].

Figures 1(b) and (c) depict how to compute statistical estimates for $E[Y_t]$ and $E[Y]$. Such estimates can and should be accompanied by appropriate measure of uncertainty, e.g., computing "α-δ confidence intervals" (CI) around them. Given two user-specified parameters $\alpha \in (0, 1)$ and $\delta \in \mathbb{R}^+$, we will show how to guarantee with statistical confidence $(1 - \alpha) \cdot 100\%$ that the actual expected value belongs to the interval of width δ centred at its estimate, and how to optimize the number of runs needed to obtain such guarantee. These steps, which are sometimes overlooked in the ABM community, can make the statistical analysis of any stochastic simulation model sounder and more informative for policy analysis. We now provide more details on our proposals for transient and steady state analyses.

Transient analysis. Procedures for transient analysis are well-established and relatively simple. As shown in Figure 1(b), for a given time of interest t (a column) we obtain a natural (unbiased) estimator for $E[Y_t]$ by computing the *vertical* mean $\overline{Y}_t$ of the observations at t (across the rows). Since these observations are IID, we can use standard statistical techniques based on the law of large numbers to build CIs as follows (see Chapter 9 of Law and Kelton, 2015):

$$\overline{Y}_t \pm \mathbf{t}_{n-1,1-\frac{\alpha}{2}} \cdot \sqrt{\frac{s_t^2}{n}}, \tag{1}$$

where n is the number of simulations, s_t^2 is the sample variance of Y_t, and the multiplier $\mathbf{t}_{n-1,1-\frac{\alpha}{2}}$ is obtained from the tabulation of the Student's T distribution with $n-1$ degrees of freedom (the area under the density function integrated from minus infinity to $\mathbf{t}_{n-1,1-\frac{\alpha}{2}}$ is equal $1 - \frac{\alpha}{2}$). For any fixed confidence level α, the width of the CI decreases as n increases. Therefore, in an automated procedure for computing an α-δ CI, we can continue performing new simulations until the width becomes smaller than the desired δ (the target width can also be expressed as a fraction of the mean value; $\delta\%$ of $\overline{Y}_t$). Note that the CI width shrinks slowly, at the rate of the square root of n. Therefore, it is important to perform the *correct* number of simulations to guarantee the target width without performing unnecessary

[6] We leave to future work extensions of our framework that would allow us to detect the number and nature of the statistical equilibria of a simulation model.

computations. MultiVeStA offers an automated procedure for doing so. Furthermore, since in many cases different times t might have different variances s_t^2, we account for the fact that different number of simulations might be required at each t to get CIs of homogeneous width across times. As we will se in Section 6.3, this is particularly important for counter-factual analysis.

A common exercise that builds upon transient analysis is to compare estimates obtained for different model configurations (typically corresponding to different sets of input parameter values) – as to assess whether the configurations differ significantly in terms of the output variable(s) under consideration. Given the outcomes of the transient analyses for the two configurations and a user-defined significance level a_w, our tool-box performs a *Welch's t-test* of means' equality (Welch, 1947) for every t of interest. MultiVeStA also computes the power of such test (Chow et al., 2002) in detecting a difference of at least a given (precision) ε (see Section 3.1.2) [7].

Steady state analysis. As depicted in Figure 1(c), a steady state analysis can be performed similarly to a transient analysis by adding a pre-processing step. We first compute the *horizontal* mean $\overline{Y}_i(w)$ within each simulation i, ignoring a given number of initial observations w. Since all these means are IID, we can compute their *vertical* mean $\overline{Y}(w)$ and build a CI around it as in Equation (1).

Unfortunately, this approach has intricacies that hinder its automatic implementation and can lead to relevant analysis errors. Depending on the chosen number w of initial observations to discard, the estimator $\overline{Y}(w)$ of $E[Y]$ might carry a bias due to the transient behavior of the system, and not give us reliable information on its steady state (see Section 7.2 for a notable example from the literature). In order to avoid this issue, we need to identify the *correct* w the system needs to exit its transient (or *warmup*) period, and discard the initial w observations from each simulation. Such procedure is known as *Replication and Deletion* (RD, Law and Kelton, 2015). Effectively identifying the length of the warmup period is a difficult problem. The most popular approaches in the ABM community are rooted in the Welch's method (Welch, 1983):

1. Perform n simulations of given length m and compute averages $\overline{Y}_t$, $t = 1, \ldots, m$ as in Figure 1(b);
2. Plot $\overline{Y}_t$, $t = 1, \ldots, m$ [8];
3. Choose the time w after which the plot *seems to converge*. If no such time exists, iterate the procedure from point (1), performing a new batch of n simulations of length m, and computing averages over all simulations.

Being only semi-automated and based on a visual assessment, this procedure is time consuming, error-prone, and not backed by a strong statistical justification. It also critically depends on choosing a large enough "time horizon" m – of course progressively larger m can be tried, adding to the computational burden.

More recently, Grazzini (2012) presented an alternative approach where a single simulation of length m is performed and divided into *windows* of length wi (m and wi are arbitrarily chosen). If the distribution of the means computed within each window passes a randomness test (in particular the Runs Test by Gibbons, 1986; Wald and Wolfowitz, 1940), then the author concludes that the system is in steady state. The use of statistical tests rather than visual assessments makes the approach more reliable, fostering its use in the ABM literature (e.g., Guerini and Moneta, 2017; Lamperti et al., 2020). However, the approach is still not fully automated – and relies on the arbitrary choice of m and wi; quoting from the author "*with appropriate settings the tests can detect non-stationarity*" (Grazzini, 2012). In the next Section we introduce a fully automated statistical procedure for estimating the end of the warmup period.

[7] In Section 6.3 we show that a reasonable choice is to set $\varepsilon = \delta$.

[8] In point (2) one might smooth the plot, e.g., employing moving-windows averages, where one is in effect further averaging each $\overline{Y}_t$ with a few neighbouring steps.

3. Automated simulation-based analysis with statistical guarantees

Our approach to transient and steady state analysis is fully automated, in that all parameters are computed automatically or have default values. The user specifies the properties to be studied, the α and δ parameters to be employed in the CI construction, and an optional maximum number of allowed simulations (if this number is reached before satisfying the CI constraints, the analysis terminates with the currently computed CIs). As in Section 2, we focus the description on a single variable Y_t, but our treatment applies straightforwardly to the analysis of multiple model characteristics (indeed, MultiVeStA implements multi-variable analyses).

3.1. Transient analysis

Section 3.1.1 describes how to estimate transient properties expressed as expected values, $E[Y_t]$, and how to build CIs around them. After this, Section 3.1.2 describes how to statistically compare estimates from different experiments.

3.1.1. Mean estimation and CI computation

Algorithm 1 illustrates `autoIR`, a simple automated algorithm for transient analysis that takes in input bl (discussed later), a time of interest t, and α and δ, and produces in output an estimate of $E[Y_t]$ and a corresponding CI. The algorithm determines automatically the number n of simulations required to guarantee that the $(1-\alpha)\times 100\%$ CI centred at the estimate has width at most δ.

Lines 2-4 set t as time horizon m, and initialize the counter n of computed simulations and the list μ to store the observations at step t from each simulation (the $y_{i,t}$ in Figure 1(b)). Lines 6-11 perform a *block* of bl simulations (by default 20 (Law and Kelton, 2015)), populating μ. In Line 7, y is a list of size m containing a value $y_{i,t}$ for each time point t from 1 to m for the current simulation i, but only the value for $t = m$ is used, adding it to μ. After performing bl simulations, `autoIR` computes the mean $\overline{\mu}$ and variance s^2 of μ, used to compute the width d of the current CI. If d is greater than δ, [9] `autoIR` performs another block of bl simulations, otherwise it returns the current CI. The implementation of `autoIR` in MultiVeStA allows one to concurrently estimate $E[Y_t]$ for different time points t (e.g. average bankruptcies in each t from 1 to 400 in Section 6). This is done by computing, at each iteration, mean, variance and CI only for the elements of y (Line 7) that correspond to time points whose current CI width is still above δ. At each iteration of a block of bl simulations, the time horizon m is updated with the largest t still to be processed.

3.1.2. Test for equality of means and power computation

MultiVeStA allows one to compare, in a statistically meaningful and reliable way, expected values corresponding to different settings or parametrizations of a model. Given that the compared means might come from experiments with different sample sizes and variances, we use the Welch's t-test (Welch, 1947), whose *power* can be computed as in Chow et al. (2002).

The Welch's t-test. Given estimates from two transient analyses for a set of time points T, our tool performs a test for equality of means for each $t \in T$ using (Welch, 1947). In symbols, given two experiments $\{j,k\}$, define the set of triplets $\mathcal{D} = \{(\overline{Y}_{i,t}, s^2_{i,t}, n_{i,t}) \mid i \in \{j,k\}, t \in T\}$, each containing the mean, the sample variance, and number of simulations for time t in experiment i. MultiVeStA takes $\mathcal{D}$ as input and, for each t, computes

$$\tau_t = \frac{\overline{Y}_{j,t} - \overline{Y}_{k,t}}{\sqrt{f_{j,t} + f_{k,t}}} \ , \tag{2}$$

[9] For the sake of presentation, all algorithms in the paper consider δ given as absolute values. The case of δ given in percentage terms relatively to the studied means is trivially obtained by changing the comparisons $d > \delta$ in $d/\overline{\mu} > \delta$.

Algorithm 1 `autoIR`: *Transient analysis*

Require: bl, α, δ, t (default: 20, 0.05, 0.1, NA)

//Set t as time horizon m, and initialize data structures
$m \leftarrow t$
$n \leftarrow 0$
$\mu \leftarrow$ empty list
repeat
  for $i \in \{1, \dots, bl\}$ **do**
    $y \leftarrow$ `drawIndependentSimulation`(m)
    //Add $y_{i,t}$ *to* μ
    μ.add$(y[m])$
    $n \leftarrow n + 1$
  end for
  $(\overline{\mu}, s^2) \leftarrow$ `computeMeanAndVariance`(μ)
  $d \leftarrow$ `computeCIWidth`$(\overline{\mu}, s^2, n, \alpha)$
until $d > \delta$
return $(1 - \alpha) \cdot 100\%$ CI $[\overline{\mu} - \frac{d}{2}, \overline{\mu} + \frac{d}{2}]$ of width at most δ

Algorithm 2 `autoRD`: *Steady state analysis by Replication and Deletion*

Require: $B, b, bs, minVar, bl, \alpha, \delta$ (default: 128, 4, 16, 1E-7, 20, 0.05, 0.1)

$w \leftarrow$ `autoWarmup`$(B, b, bs, minVar)$
$m \leftarrow w \cdot 2$
$n \leftarrow 0$
$\mu \leftarrow$ empty list
repeat
  for $i \in \{1, \dots, bl\}$ **do**
    $y \leftarrow$ `drawIndependentSimulation`(m)
    $y' \leftarrow (y_{w+1}, \dots, y_m)$
    μ.add(`computeMean`(y'))
    $n \leftarrow n + 1$
  end for
  $(\overline{\mu}, s^2) \leftarrow$ `computeMeanAndVariance`(μ)
  $d \leftarrow$ `computeCIWidth`$(\overline{\mu}, s^2, n)$
until $d > \delta$
return $(1 - \alpha) \cdot 100\%$ CI $[\overline{\mu} - \frac{d}{2}, \overline{\mu} + \frac{d}{2}]$ of width at most δ

where $f_{i,t} = s^2_{i,t}/n_{i,t}$, $i \in \{j, k\}$. Following Welch (1947), under the null hypothesis that the difference between the two means is zero, each τ_t follows a Student's t-distribution with degrees of freedom approximated as in Satterthwaite (1946):

$$\nu_t \approx \frac{(f_{j,t} + f_{k,t})^2}{f^2_{j,t}/(n_{j,t} - 1) + f^2_{k,t}/(n_{k,t} - 1)} \ .$$

Therefore, given a statistical significance a_w, MultiVeStA uses τ_t to perform the test of no difference between the two means producing 1 if $\tau_t \in [-\mathbf{t}_{\nu_t, 1-\frac{a_w}{2}}, \mathbf{t}_{\nu_t, 1-\frac{a_w}{2}}]$ (the null hypothesis of equal means is not rejected) and 0 otherwise. The significance a_w is user-specified, and can be set to be equal to the α used for the transient analysis.

Power of the test. Following Chow et al. (2002), MultiVeStA estimates the power $1 - \beta_t$ of Welch's t-test in detecting a difference of at least a given precision ε between the two means at time t. This is

$$\beta_t = \mathcal{T}_{\nu_t}\left(\mathbf{t}_{\nu_t, 1-\frac{a_w}{2}} \,\middle|\, \frac{|\varepsilon|}{\sqrt{f_{j,t} + f_{k,t}}}\right) \ , \tag{3}$$

where $\mathcal{T}_{\nu_t}(x \,|\, \theta)$ is the cumulative distribution function of a non-central t-distribution with ν_t degrees of freedom and non-centrality parameter θ, evaluated at point x. Calculating the power of Welch's t-test requires specifying the minimum difference ε (Chow et al., 2002). As a rule of thumb, we suggest setting $\varepsilon \geq \delta$, the parameter used in the transient analysis, which expresses a precision for the estimated mean. In Section 6, setting $\varepsilon = \delta$ leads to very good power for the considered macro ABM.

3.2. Steady state analysis

A statistically sound analysis of steady state properties poses challenges that have been thoroughly investigated by the simulation community – at the boundary of computer science and operations research. Two main approaches have emerged (Alexopoulos and Goldsman, 2004; Whitt, 1991; Law and Kelton, 2015): those based on *Replication and Deletion* (RD; see Section 2), and those based on *batch means* (BM) (Conway, 1963; Alexopoulos and Seila, 1996; Steiger et al., 2005). Unlike RD, which computes *many short* simulations, BM computes *one long* run which is evenly divided into adjacent non-overlapping subsamples labelled as *batches*. Intuitively, if certain statistical properties hold,

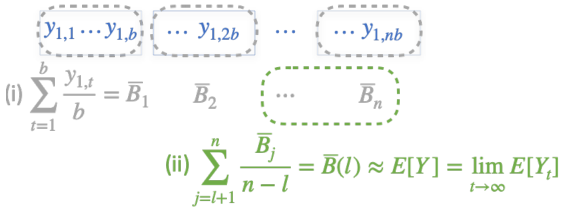


Figure 2: Steady state analysis by *Batch Means* (BM) using one long simulation: (i) We split the simulation into batches (consecutive steps) of size b, and we compute the mean within each batch (the batch means $\overline{B}_i$); (ii) We compute the mean of such means, the *grand mean*, ignoring the first l batches where it is assumed to terminate the warmup. We obtain $\overline{B}(l)$, an estimator for $E[Y]$.

each batch can be used as a simulation in RD – as depicted in Figure 2. This can be seen as a generalized version of the proposal by Grazzini (2012), which allows one to estimate the end of the warmup period rather than to check whether a given time is subsequent to such end[10].

There is no *best* approach between RD and BM (Alexopoulos and Goldsman, 2004; Whitt, 1991; Kelton and Law, 1984). They are complementary, and therefore have complementary (dis)advantages. RD, which uses many short simulations, suffers from biases due to initial conditions. BM, which uses many short batches from one long simulation, is less affected by initialisation bias but suffers from correlations among batch means.

While some automated BM-based procedures have been proposed (e.g., Steiger et al., 2005; Tafazzoli et al., 2011; Gilmore et al., 2017), to the best of our knowledge, little attention has been paid to RD. Interestingly, Lada et al. (2013) tried to combine the two approaches exploiting their respective strengths: they use BM for warmup analysis, and automate the use of RD by discarding the estimated transient behaviour from each simulation. Following a similar approach, we extract and condense the warmup analysis capabilities inspired by BM into a simple self-standing procedure for warmup estimation (`autoWarmup`), and we introduce automated RD- and BM-based algorithms (`autoRD` and `autoBM`, respectively) which use `autoWarmup`. In all algorithms (see Figure 3) we favour simplicity and accessibility.

3.2.1. Steady state analysis by replication and deletion

Algorithm 2 illustrates `autoRD`. The *difficult part* in automating RD is the warmup analysis. However, in our setting we can easily do this by invoking `autoWarmup` (Line 1; see Section 3.2.2 below). For now it is sufficient to know that w is the last step of the estimated warmup period. Once w has been determined, we have to set a *substantially larger* time horizon (Law and Kelton, 2015). We can do this using a (small) multiplier. In Line 2 the default multiplier for w is 2. The code of `autoRD` presents also a second modification with respect to that of `autoIR`: we replaced Line 9 of Algorithm 1 with Lines 8-9 of Algorithm 2 to discard the first w observations from y, and add the mean of the remaining values of y (the horizontal mean in Figure 1(b)) to μ.

3.2.2. Warmup estimation

Algorithm 3 provides pseudo-code for our automatic warmup estimation, inspired by existing BM-based approaches for steady state analysis (Steiger et al., 2005; Gilmore et al., 2017; Tafazzoli et al., 2011). Indeed, such algorithms include a form of warmup analysis that we extract and refine into a simple self-standing procedure. Lines 1-5

[10]More precisely, Grazzini (2012) appears to employ a non-automated version of BM. Yet the first automated version of BM was published in 1979 (Law and Carson, 1979). This is a clear signal of the potential (and often overlooked) complementaries between the simulation community and the ABM community in economics.

Algorithm 3 `autoWarmup`: *Warmup estimation*

Require: $B, b, bs, minVar$, (default: 128, 4, 16, 1E-7)

1: *//Draw the first $B \cdot bs$ steps*
2: $\mu \leftarrow$ `array`(B)
3: **for** $i \in \{1, \ldots, B\}$ **do**
4: $\quad \mu[i] \leftarrow$ `drawBatchAndComputeMean`(bs)
5: **end for**
6: $(a, \rho) \leftarrow$ `goodnessOfFitTests`$(\mu, b, minVar)$
7: *//Keep doubling bs and time horizon until tests pass*
8: **while** $a > a^*$ `or` $\rho > \rho^*$ **do**
9: $\quad$ **for** $i \in \{1, \ldots, B/2\}$ **do**
10: $\quad\quad \mu[i] \leftarrow (\mu[2 \cdot i] + \mu[2 \cdot i + 1])/2$
11: $\quad$ **end for**
12: $\quad bs \leftarrow 2 \cdot bs$
13: $\quad$ **for** $i \in \{B/2 + 1, \ldots, B\}$ **do**
14: $\quad\quad \mu[i] \leftarrow$ `drawBatchAndComputeMean`(bs)
15: $\quad$ **end for**
16: $\quad (a, \rho) \leftarrow$ `goodnessOfFitTests`$(\mu, b, minVar)$
17: **end while**
18: **return** Warmup period estimated to terminated after $B \cdot bs$ steps

Algorithm 4 `goodnessOfFitTests`

Require: $\mu, b, minVar$

1: $\mu' \leftarrow (\mu_{b+1}, \ldots, \mu_B)$
2: $(\overline{\mu}, s^2) \leftarrow$ `computeMeanAndVariance`(μ')
3: $(\alpha, \rho) \leftarrow (0, 0)$
4: **if** $s^2 > minVar$ **then**
5: $\quad a \leftarrow$ `AndersonDarlingNormalityTest`$(\mu', \overline{\mu}, s^2)$
6: $\quad \rho \leftarrow$ `lag1Autocorrelation`$(\mu', \overline{\mu}, s^2)$
7: **end if**
8: **return** (a, ρ);

Algorithm 5 `autoBM`: *Steady state analysis by Batch Means*

Require: $B, b, bs, minVar, \alpha, \delta$ (default: 128, 4, 16, 1E-7, 0.05, 0.1)

1: `autoWarmup`$(B, b, bs, minVar)$ *//Fast-forward simulation after warmup*
2: $\mu \leftarrow$ `array`(B)
3: **for** $i \in \{1, \ldots, B\}$ **do**
4: $\quad \mu[i] \leftarrow$ `drawBatchAndComputeMean`(bs)
5: **end for**
6: $(a, \rho, \overline{\mu}, d) \leftarrow$ `goodnessOfFitTestsAndCI`$(\mu, b, minVar, \alpha)$
7: *//Keep doubling bs and time horizon until tests pass*
8: **while** $a > a^*$ `or` $\rho > \rho^*$ `or` $d > \delta$ **do**
9: $\quad$ **for** $i \in \{1, \ldots, B/2\}$ **do**
10: $\quad\quad \mu[i] \leftarrow (\mu[2 \cdot i] + \mu[2 \cdot i + 1])/2$
11: $\quad$ **end for**
12: $\quad bs \leftarrow 2 \cdot bs$
13: $\quad$ **for** $i \in \{B/2 + 1, \ldots, B\}$ **do**
14: $\quad\quad \mu[i] \leftarrow$ `drawBatchAndComputeMean`(bs)
15: $\quad$ **end for**
16: $\quad (a, \rho, \overline{\mu}, d) \leftarrow$ `goodnessOfFitTestsAndCI`$(\mu, b, minVar, \alpha)$
17: **end while**
18: **return** $(1 - \alpha) \cdot 100\%$ confidence interval $[\overline{\mu} - \frac{d}{2}, \overline{\mu} + \frac{d}{2}]$ of width at most δ, adjusted for keeping into account residual correlation

Algorithm 6 `goodnessOfFitTestsAndCI`

Require: $\mu, b, minVar, \alpha$

1: $\mu' \leftarrow (\mu_{b+1}, \ldots, \mu_B)$
2: $(\overline{\mu}, s^2) \leftarrow$ `computeMeanAndVariance`(μ')
3: $(\alpha, \rho, d, n) \leftarrow (0, 0, 0, B - b)$
4: **if** $s^2 > minVar$ **then**
5: $\quad a \leftarrow$ `AndersonDarlingNormalityTest`$(\mu', \overline{\mu}, s^2)$
6: $\quad \rho \leftarrow$ `lag1Autocorrelation`$(\mu', \overline{\mu}, s^2)$
7: $\quad d \leftarrow$ `computeCIWidth`$(\overline{\mu}, s^2, n, \alpha, \rho)$
8: **end if**
9: **return** $(a, \rho, \overline{\mu}, d)$;

Figure 3: BM-based algorithms for estimating the initial warmup period (left), and for studying steady state properties (right).

perform a simulation of $m = B \times bs$ steps (by default, $B = 128$ and $bs = 16$). The simulation is divided in B adjacent non-overlapping *batches*, each containing bs steps. After this, the array μ stores the mean of each batch (therefore the name *batch means*): each entry $\mu[i]$ stores the corresponding $\overline{B}_i$ (see Figure 2). The algorithm then proceeds iteratively by performing statistical tests to check whether m is large enough to cover the warmup period, doubling the number of performed steps while keeping the number of batches fixed (doubling the steps bs in each batch) until all tests are passed. The key point is that if the process satisfies properties required for steady state analysis (Law and Carson (1979); Steiger and Wilson (2001); see also Section 4), then such iterative procedure will lead to *approximately* IID normally distributed batch means μ for a sufficiently large value of bs.

BM-based approaches perform different statistical tests on μ to check whether m is large enough for completing the warmup period: Tafazzoli et al. (2011) use the von Neumann (1941) randomness test, while Steiger et al. (2005) use a test for stationary multivariate normality on groups of 4 consecutive batches followed by a check for low correlation among consecutive batch means (i.e., the lag-1 autocorrelation of μ). Gilmore et al. (2017) apply the Anderson-Darling test for normality on μ, followed by a check for low lag-1 autocorrelation of μ. In all cases, a few (typically 4) initial batches are ignored as they are likely the most affected ones by the initial transient. We follow the latter approach, as specified in the subprocedure `goodnessOfFitTests` of Algorithm 4: Line 1 skips b (by default 4) initial batches, obtaining μ', then Line 2 computes the variance of the batch means, used in Line 4 to decide whether the statistical tests are necessary or pass by default. The rationale is that if the variance among the batch means is below a minimum threshold (parameter *minVar* with default value 1E-7), then the process is likely converging to a deterministic fixed point, therefore we can safely assume that the intial warmup period has terminated. Concerning normality, Line 5 uses the Anderson-Darling test implemented in the SSJ library (L'Ecuyer, 2016; L'Ecuyer et al., 2002) to check whether it is statistically plausible that μ' has been sampled from a normal distribution specified by its mean and variance, and obtain a p-value a. Line 6 stores ρ, the lag1-autocorrelation of μ'. The subprocedure thus returns a and ρ, which are used in Line 8 of `autoWarmup` to decide whether the tests are passed – using minimum thresholds a^* and ρ^* based on prior publications (Gilmore et al., 2017; Steiger et al., 2005) [11]. If any of the tests fail, then an iteration of the *while* loop in Lines 8-17 is performed to double the number of steps m by doubling the current batch size bs. We note that the current B batch means are *squeezed* in the first half of μ, (Lines 9-11), and m new steps are performed to create the new batch means in the second half of μ (Lines 13-15). The statistical tests are performed on the new batch means, and new iterations of the loop are performed until both statistical tests are passes. The algorithm terminates returning the final value of $m = B \times bs$ as the estimated end of the warmup period.

3.2.3. Steady state analysis by batch means

Algorithm 5 illustrates our automatic BM-based procedure for steady state analysis. Line 1 invokes `autoWarmup`, which *moves* the simulator to the end of the estimated warmup period. No other information from `autoWarmup` is used. The algorithm then proceeds similarly to `autoWarmup`, the only difference being that we add a third statistical test: we also compute the width d of the CI according to the current batch means. This is obtained by invoking `goodnessOfFitTestsAndCI` from Algorithm 6 rather than `goodnessOfFitTests` Since the tests for normality and absence of correlation were already passed during `autoWarmup`, one might expect that they are no longer necessary. We note however that the tests passed for the last value of bs used in `autoWarmup`, so they could potentially fail for initial small values of bs. When all statistical tests are passed, we return the computed $(1-\alpha)100\%$ CI of width at most δ, adjusted by the computed residual correlation among the batch means. This is done similarly to Steiger et al. (2005), using an inverse Cornish-Fisher expansion (Stuart and Ord, 1994) based on a standard normal density [12].

3.2.4. Some remarks on `autoBM` *and* `autoRD`

We note that both `autoBM` and `autoRD` proceed by iterations, during which new samples are drawn and new statistical tests are performed. In `autoBM`, new simulation steps are added onto the same long simulation (the number of simulations n is constant, the time horizon m grows). In `autoRD`, new simulations of fixed length are added (n grows, m is constant). In some sense, the computational burden of `autoRD` is higher, as w steps from each newly performed simulation are ignored. However, the simulations performed in each iteration of `autoRD` can be trivially parallelized – so the additional computation can be efficiently handled. Rather, which approach to prefer depends on the model at hand and on the available hardware:

- The longer the initial warmup period, the more advantageous is `autoBM` relative to `autoRD`.
- The larger is the degree of parallelism supported by the hardware, the more advantageous is `autoRD` relative to `autoBM`.

[11] In particular, the significance level for the normality test is set to $a^* = 1\%$ while the lag-1 autocorrelation threshold is set to $\rho^* = \sin(0.927 - \frac{q}{\sqrt{size(\mu)}})$, where q is the 99% quantile of the standard normal distribution. See (Steiger et al., 2005) for more details.

[12] See Section 2 of (Steiger et al., 2005) for the exact formula.

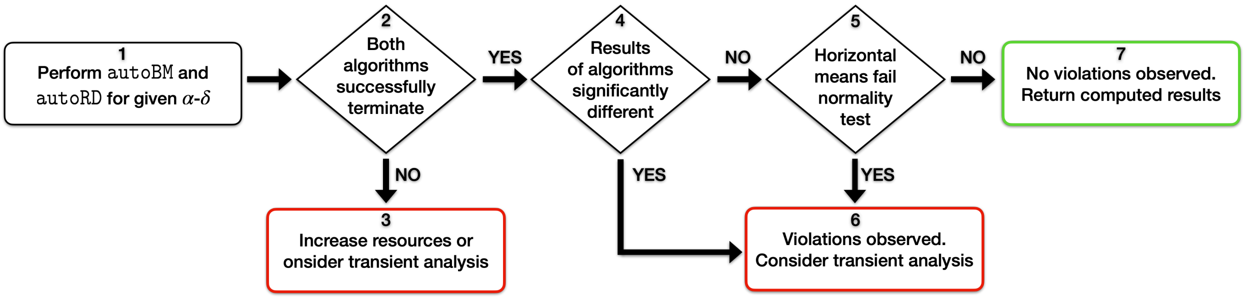


Figure 4: Procedure for ergodicity diagnostics based on `autoRD` and `autoBM` to asses the reliability of a steady state analysis

The ABM community favours the RD approach due to its simplicity, but its *trivially parallelizable* nature is not always exploited due to limited computer engineering skills. Notably, some studies have shown that the BM approach might provide more accurate results in specific cases (Whitt, 1991; Alexopoulos and Goldsman, 2004). The implementation of both algorithms in MultiVeStA enables modelers to freely choose between the two and to exploit the distributed nature of the tool-box to parallelize simulations. The next section shows how the RD and BM can be combined to obtain a methodology for ergodicity diagnosis.

4. Ergodicity diagnostics and detection of multiple stationary points using `autoRD` and `autoBM`

Consistency and unbiasedness of the estimates produced by `autoBM` and `autoRD` rely on the underlying process possessing the *strong mixing* property [13]. Indeed, once normality of batch means in `autoWarmup` is well approximated and autocorrelation is low, we can be confident that future observations will not have initialization bias (Steiger et al., 2005). If at a certain point in time the batch means resemble a sample of IID observations from the same Gaussian population, then the non-random effects of initial conditions must have disappeared. Further, the strong mixing assumption ensures that such a point in time can eventually be reached by increasing the batch size (Law and Carson, 1979; Steiger and Wilson, 2001). Here we describe a procedure that combines `autoRD` and `autoBM` to assess whether this assumption is met. The procedure, depicted in Figure 4, is fully implemented in MultiVeStA and is validated in Section 8 on variants of a prediction market model.

We start performing both `autoBM` and `autoRD` for given α and δ (step 1). If any of the two fails to converge in due time (step 2), we have evidence that the process is eventually non-stationary (or fails to reach its stationary phase within the allotted computational time/resources). In such cases, performing any steady state analysis could be misleading and should be avoided (step 3).

If both `autoBM` and `autoRD` successfully terminate, we can be confident that the process possesses *ergodic properties* (Gray, 2009; Billingsley, 1995) – meaning, intuitively, that the horizontal means of its realisations (i.e., the means across simulations as in Fig. 1(c)) indeed converge asymptotically to a finite number. However, there could be potentially different limits for different simulations. In these cases, a natural check for ergodicity is to compare the results of `autoBM` and `autoRD` (step 4). This is in line with previous approaches to ergodicity analysis from the literature (e.g., Grazzini, 2012), where BM-like means across one long simulation are compared with RD-like means across several simulations. The difference is that our BM and RD results are obtained using automated algorithms (`autoBM`

[13] The strong mixing property guarantees that two sufficiently distant observations in $(\mathbf{Y}_t)_{t>0}$ are approximately independent. There are various definitions of the property; we utilize the ϕ-mixing definition provided in Steiger and Wilson (2001)

and `autoRD`), rather then from arbitrarily parametrized experiments. Our test, performed in step 4, checks whether the difference between BM and RD estimates is larger in absolute value than the δ used for CI implementation. If this is the case, we have evidence of non-ergodicity and therefore of violation of the strong mixing assumption (step 6). For example, this could be due to the presence of multiple stationary points in the process: `autoBM` would end up exploring only one of such stationary points, while `autoRD` would provide averaged information on the possible realizations. If the difference between BM and RD estimates is small, we have no evidence that the assumption is violated and we proceed with a second test on `autoRD`'s horizontal means (step 5). Indeed, under the null hypothesis that the process is strongly mixing and that the initial warmup phase has been effectively discarded, the central limit theorem for weakly correlated variables states that the horizontal means should be approximately normally distributed. In particular, we perform an Anderson-Darling normality test (with significance level 1%) on the sample of horizontal means. If the null hypothesis is not rejected we again have no evidence of violation of the strong mixing assumption, and we therefore return the values computed by either of the two algorithms (step 7).

5. Operationalizing the framework: Statistical Model Checking and MultiVeStA

This section discusses how we operationalise our approach. In particular, we frame our approach to ABM analysis in the context of Statistical Model Checking and show how we integrate it into MultiVeStA, a model-agnostic statistical model checker that can be integrated with existing simulators.

5.1. Statistical Model Checking

Statistical Model-Checking (SMC) (Agha and Palmskog, 2018; Legay et al., 2019) is a successful simulation-based verification approach from computer science. SMC allows to study quantitative properties of large-scale models through completely automated analysis procedures equipped with statistical guarantees. Following the principle of *separation of concerns*, the idea is to offer a simple external language to express properties of interest that can be *queried* on the model using predefined analysis procedures. The goal of SMC is therefore that of offering a *one-click-analysis* experience to the modeler which is freed from the burden of modifying the model to generate large CSV files every time a new analysis is required, and then analyzing such CSV files in an error-prone semi-automated manner. This guarantees that the analysis procedures are written once and then extensively tested, decreasing the possibility of errors. Making a parallel with databases, we do not have to explicitly manipulate the internal representation of the data every time a new query is needed, rather we define the data to be selected using compact languages (e.g. SQL).

Several statistical model checkers exist, most of which require to implement models into proprietary languages. We consider *black-box* SMC (Sen et al., 2004; Younes, 2005), where the idea is to offer a model-independent analysis framework that can be easily attached to existing simulation models, effectively enriching them with automated statistical analysis techniques. In particular, we use MultiVeStA (Sebastio and Vandin, 2013; Gilmore et al., 2017), redesigned and extended here with the techniques from the previous sections, tailored for ABM community.

5.2. Simulator integration

MultiVeStA only needs to interact with a simulator by triggering 3 basic actions: (*i*) `reset(seed)`, to reset the simulator to its "initial state", and update the random seed used to generate pseudo-random numbers. This is necessary to reset the model before performing a new simulation; (*ii*) `next`, to perform one step of simulation; (*iii*) `eval(obs)`, to evaluate an observation in the current simulation state, where an observation (`obs`) can be any feature of the aggregate model or of any group of agents. A new model can be integrated with MultiVeStA by implementing an *adaptor* between MultiVeStA and the considered simulator, obtained by instantiating MultiVeStA's (Java) interface. As a consequence, it natively supports Java-based simulators, but it has been also integrated with C- and Python-based simulators, and it has been recently extended to support R-based ones.

For the ABM macro model from Section 6 we are interested in two aggregate features of the model: the number of bankruptcies and the unemployment rate in a given step. Therefore, the model has been integrated such that these can be obtained using `eval("bankruptcy")`, and `eval("unemploymentRate")`, respectively. Instead, the prediction market models from Sections 7 and 8 have been integrated such that `eval(i)` gives a particular feature of agent i (its current wealth), and `eval("price")` gives a certain aggregate feature of the model (the prevailing price).

5.3. *MultiVeStA query language and supported analysis*

MultiVeStA offers a powerful and flexible *property specification language*, MultiQuaTEx, which allows to express transient and steady state properties, including warmup analysis.

Transient properties. Intuitively, a MultiQuaTEx query might describe a random variable (e.g., the number of bankruptcies in an ABM macro model at a certain point in time during a simulation). Following the discussion in Section 2, the expected value of a MultiQuaTEx query is estimated as the mean $\overline{x}$ of n samples (taken from n simulations), with n large enough (but minimal) to guarantee that the $(1-\alpha)\cdot 100\%$ CI centred on $\overline{x}$ has size at most δ, for given α and δ.

MultiQuaTEx actually allows to express more random variables in one query, all analysed independently reusing the same simulations. Listing 1 depicts a MultiQuaTEx query used in Section 6 to study the evolution of the number of bankruptcies and of the unemployment rate in an ABM macro model.

Coming to the structure of a MultiQuaTEx query, it contains a list of *parametric operators* that can be used in an `eval autoIR` command to specify the properties to be estimated. Lines 1-4 of Listing 1 define the parametric operator `obsAtStep` having two parameters, `t` and `obs`, respectively the step and observation of interest. Such operator is evaluated, in every simulation, as the value of `obs` at time point `t`. Before discussing the *body* of the operator, we note that Line 5 uses it twice for observations the number of bankruptcies and the unemployment rate for each step from 1 to 400 (with increment 1). Therefore 800 properties will be studied (400 for each observation), all evaluated using the same simulations and with their own CI. The body of an operator (Lines 1-4) might contain:

1. conditional statements (the `if-then-else-fi`);
2. real-valued observations on the current simulation state (the `s.eval` in Line 1 and Line 2);
3. a `next` operator that triggers the execution of a simulation step (Line 3);
4. recursion, used in Line 3 to evaluate `obsAtStep(t,obs)` in the next simulation step;
5. arithmetic expressions.

This is general enough to express a wide family of properties at varying of time. In the case of Listing 1, we check whether we have reached the step of interest (Line 1), in which case we return the required observation (Line 2). Otherwise, we perform a step of simulation (Line 3), and evaluate recursively the operator in the next simulation state.

```
1 obsAtStep(t,obs) = if (s.eval("steps") == t)
2                        then s.eval(obs)
3                        else next(obsAtStep(t,obs))
4                    fi ;
5 eval autoIR(E[ obsAtStep(t,"bankruptcy") ],E[ obsAtStep(t,"unemploymentRate") ],t,1,1,400) ;
```

Listing 1: A transient MultiQuaTEx query

```
1 obs(o) = s.eval(o) ;
2 eval warmup(E[ obs(0) ],E[ obs(1) ],E[ obs(2) ],E[ obs("price") ]) ;
3 eval autoBM(E[ obs(0) ],E[ obs(1) ],E[ obs(2) ],E[ obs("price") ]) ;
4 eval autoRD(E[ obs(0) ],E[ obs(1) ],E[ obs(2) ],E[ obs("price") ]) ;
```

Listing 2: A steady state MultiQuaTEx query. Only one of the three eval commands should be used at a time.

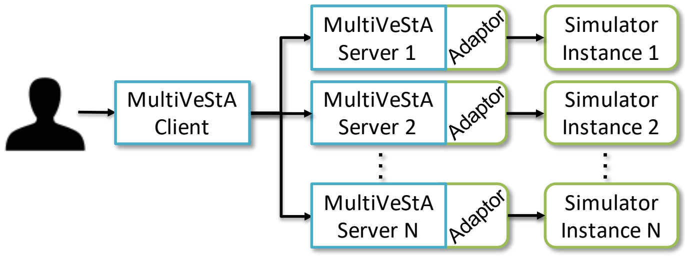


Figure 5: MultiVeStA's client-server architecture enabling parallelization of simulations.

Steady state properties and warmup analysis. MultiQuaTEx has been extented to support MultiVeStA's extension with steady state and warmup analysis capabilities discussed in Section 3.2. Listing 2 provides a *steady state* MultiQuaTEx query used in Section 7 to study the average value at steady state of the wealth of three agents (0, 1, and 2), and of the price in our test-bed market selection model. The query is simple, as in this case the operator `obs` just returns the observation of interest, while Lines 2-4 show how to run the three types of supported analysis. In particular, a steady state query is composed of two parts: A list of `next`-free operators, and one of the three `eval` commands in Listing 2, provided with a list of operators to study.

Intuitively, a steady state MultiQuaTEx query defines observations on single simulation states, implicitly studied at steady state. In particular, `warmup` performs the warmup estimation procedure (Section 3.2.2) for each of the listed properties. Indeed, every random variable defined on a process might have a different warmup period. We will see examples of this in Section 7. Instead, `autoBM` performs a warmup estimation on each property, and, begins computing the batch means procedure (Section 3.2.3) on each of them as soon as the property completes its warmup period. The command `autoRD` is similar, but it first completes the warmup analysis for all considered properties, and then feeds this information to the replication deletion procedure from Section 3.2.1. In all cases, the default values described in Section 3 will be used if not otherwise specified by the user when running the analysis.

MultiQuaTEx supports two further `eval` commands: `manualBM` and `manualRD`. These behave the same as `autoBM` and `autoRD`, respectively, but skip the warmup analysis phase and required as input an estimation of the warmup period. These might be useful in case one has this information due to previous analyses. In Section 7 we use them to replicate erroneous steady state analyses from the literature based on a wrong estimation of the warmup period.

5.4. *MultiVeStA's distributed architecture*

MultiVeStA has a client-server architecture as sketched in Figure 5. This is a classic software architecture for distributing tasks in the cores of a machine or in the nodes of a network. We distribute the simulations of `autoIR` and `autoRD`. In the figure, arrows denote visibility/control/activation of the source component on the target one:

- A user runs the client specifying the model, query, CI, and the parallelism degree N. Transparently to the user, the client will trigger, distribute, and handle the necessary simulations providing to the user the results.
- The client creates N servers among whom distributes the analysis tasks.

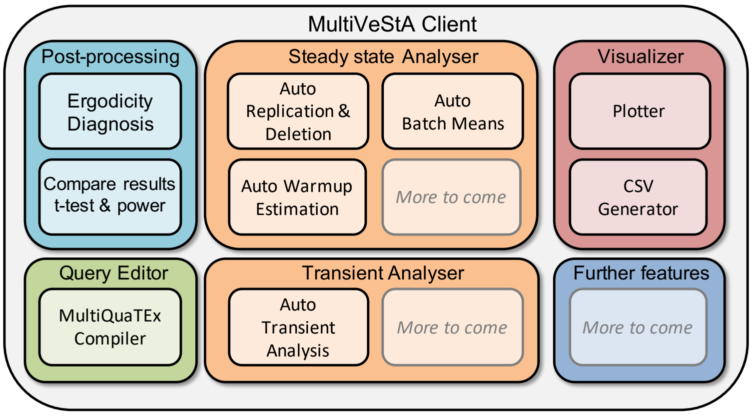


Figure 6: The novel architecture of the MultiVeStA client

- Each server runs independently, therefore in parallel, the required simulations. Each server creates its own instance of the simulator, and controls it through the adaptor to perform the simulations.

As discussed, we extended MultiVeStA with a number of analysis techniques. In particular, we mainly extended the client, where the analysis logic is localized. The new architecture of the client is depicted in Figure 6. It consists of a number of modules, the central ones regarding steady state and transient analysis. Further modules regard: post-processing of analysis computed by MultiVeStA like t-tests and power computation to compare results obtained for different model configurations (Section 3.1.2), or the methodology for ergodicity analysis (Section 4); support for the creation and parsing of MultiQuaTEx queries, offered by a novel compiler for MultiQuaTEx queries; visualization of the analysis results through a plotter and of a CSV file creator.

6. Application: Transient analysis of a large macro ABM

We apply the transient analysis from Section 3.1 to the large-scale macro-financial ABM of Caiani et al. (2016).

6.1. The macro ABM of Caiani et al. (2016)

The model has been developed to bridge the stock flow consistent approach (SFC; Godley and Lavoie (2006)) with the macroeconomic agent based literature (see, e.g., Delli Gatti et al., 2005; Cincotti et al., 2010; Dosi et al., 2010; Dawid et al., 2012; Popoyan et al., 2020).[14] It depicts an economy composed of households selling their labor to firms in exchange for wages, consuming, and saving into deposits at (commercial) banks. Households own shares of firms and banks in proportion to their wealth, and receive a share of firms' and banks' profits as dividends; they also pay taxes as set by the Government, which runs fiscal policy. There are two categories of firms. Consumption firms produce a homogeneous good using labor and the capital goods manufactured by the other class of firms: capital firms. Firms may apply for loans in order to finance production and investment. Retained profits enter the financial system as banks' deposits. Banks provide credit to firms, buy bonds issued by the Government and need to satisfy mandatory capital and liquidity ratios. Finally, a Central Bank holds banks' reserve accounts and the government account, accommodates banks' demand for cash advances at a fixed discount rate, and possibly buy government bonds that have not been purchased by banks.

[14] A rather detailed overview of the macro ABM literature can be found in Fagiolo and Roventini (2017) and Dosi and Roventini (2019).

Here we focus on two key indicators of economic activity: the unemployment rate, and the bankruptcy rate of business firms. They have been chosen because of: (i) the relative large fluctuations they exhibit during the transient dynamics (see Figure 2 in Caiani et al., 2016), which we aim at reproducing and testing and (ii) their well known role as proxies of an economy's health at macro e micro level, respectively. We sketch how these two quantities are modelled by Caiani and co-authors while leaving the additional details about the model to the original paper.[15]

The labour market is composed of workers, firms, and the public sector. Firms in the capital good sector (indexed by k) demand workers based on their desired level of production y_{xt}^D and the productivity of labor (μ_N), which is assumed to be constant and exogenous:

$$N_{kt}^D = \frac{y_{xt}^D}{\mu_N}. \tag{4}$$

Differently, the request of workers by consumption good firms (indexed by c) is given by

$$N_{ct}^D = u_{ct}^D \frac{\kappa_{ct}}{l_k}, \tag{5}$$

where κ_{ct} is the capital stock, l_k is a constant expressing the capital-to-labor ratio and u_{ct} is the utilization capacity needed to obtain the desired production. Workers can be fired under two circumstances: workers in excess of production needs are randomly sampled from the pool of firm employees and fired, and workers can loose their job because of an exogenous positive employee turnover (a fixed share of workers is fired in every period). Finally, a constant share of households are employed by the public sector and public servants are also subject to an exogenous turnover.

After having planned production, firms and the government interact with unemployed households on the labor market. Workers follow an adaptive heuristic to set the wage they ask for: if over the year (i.e., four periods), they have been unemployed for more than two quarters, they lower the asked wage by a stochastic amount. In the opposite case, they increase their asked wage. The share of workers that is not employed at the end of each session of interaction in the labour market represents the prevailing unemployment rate.

After production, firms sell their products and need to compensate for the inputs they received. Firms may default when they run out of liquidity to pay wages or to honour the debt service. Defaulted firms are bailed-in by households (who are the owners of firms and banks and receive dividends) and depositors, as the authors seek to maintain the number of firms constant. Hence, the bankruptcy rate emerges as the ratio between defaulted firms before the bailing-in event and the total number of firms in the economy. As the defaulted firms create non-performing loans that might trigger vicious cycles and - ultimately - a financial crisis, they offer key information on the turbulence and riskiness of the business cycle.

6.2. Transient analysis with `autoIR`*: automatic computation of confidence intervals*

The model is run in its baseline configuration considered in Section 5.1 of Caiani et al. (2016). The artificial time series show the model first experiences a sequence of expansionary and recession regimes, then converging, in most cases, to a relatively stable behaviour where aggregate variables (including the unemployment and the bankruptcy rates) fluctuate around particular values, and nominal aggregates grow at similar rates. Our focus is centred on the first part of such process.

As a first exercise, we reproduce the behaviour of the economic indicators we selected in the first 400 steps of the simulation, and construct CIs around their mean according to Equation (1) (Figure 7). In particular, we choose

[15] Of course, all variables present in the original model could be analysed using the very same procedure; we selected two for illustrative purposes.

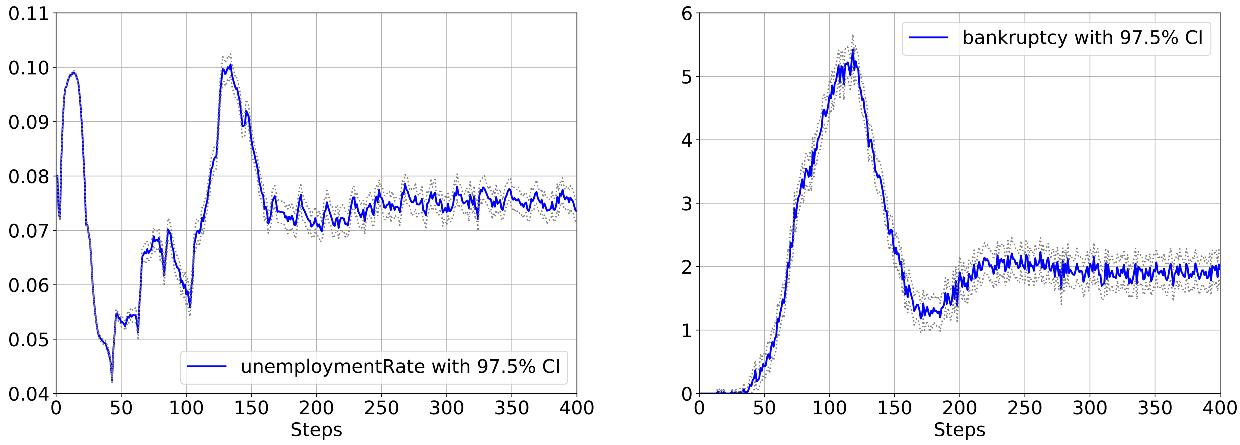


Figure 7: Unemployment rate and bankruptcy over time. The dashed lines are the computed 97.5% CI of size δ at most 0.005 and 0.5, respectively.

$\alpha = 0.025$ and set δ at the maximal allowed width of the confidence intervals around our central estimates (i.e. $\delta_U = 0.005$ for the unemployment rates and $\delta_B = 0.5$ for the average bankruptcies) and let `autoIR` automatically decide the number of simulations needed to obtain the desired confidence intervals. We stress that MultiVeStA automatically determines the number of runs required to obtain the desired CI for each point in time and for whatever variable of interest. As shown in the top of Figure 8, this required at most 378 simulations for both properties.

As a concept-proof of our approach, the inspection of Figure 7 confirms that our algorithms do not modify the model and deliver the same dynamics (see Figure 2 of Caiani et al., 2016).[16] However, Caiani et al. (2016) performed 100 simulations for all the 400 time steps, without providing information on the obtained confidence intervals.

The ability to specify the precision of the confidence intervals comes with a number of advantages. First, it is a flexible requirement that can be expressed either in absolute or relative terms (see Section 3), leaving the chance to statistically compare the expected behaviour of the model to a certain target (say, an employment rate not higher that 5% or an inflation rate of 2%) or to its mean (e.g. allowing one to compute for each period the probability to observe bankruptcy rates 10% higher than the average). Second, and more relevantly, it allows evaluating the robustness (and the uncertainty) of the dynamics simulated by the model. In particular, Figure 8 shows how the width of the confidence intervals vary, for each time point and property, across the simulation span for various number of simulations. The top of the figure shows the intermediate CI widths obtained after every iteration of the blocks of simulations performed by MultiVeStA (see the discussion in Section 3.1.1 - we use $bl = 42$). We note that the widths decrease at every iteration, and that some time points (from 100 to 200) require more simulations than the others to get the desired CI width. Instead, the bottom of the figure compares the CIs obtained by MultiVeStA (in green) against those obtained using the setting of the original paper (i.e. 100 simulations for all time steps, in red). We note that, apart for the first time points which present very low variance, the CIs computed by MultiVeStA tend to be homogeneous and close to the required δ, demonstrating that the minimum number of simulations are computed for the given δ. Instead, the setting used by Caiani et al. (2016) might lead to CIs of different widths which follow the trend of the computed means. This is particularly evident for the case of firms' bankruptcies, cf. Figure 8 (bottom-right), while the same does not happen for unemployment rates (bottom-left of the figure). The figure suggests that each property

[16] We highlight that the artificial time series we generate are somehow comparable to Figure 2 of Caiani et al. (2016); however, Caiani et al. apply a bandpass filter over their series and just show the emerging trend component. Contrarily, we show the "raw" series that the model generates. We notice that the latter is the prevailing practice in the literature (see for example the models reviewed in Fagiolo and Roventini, 2017).

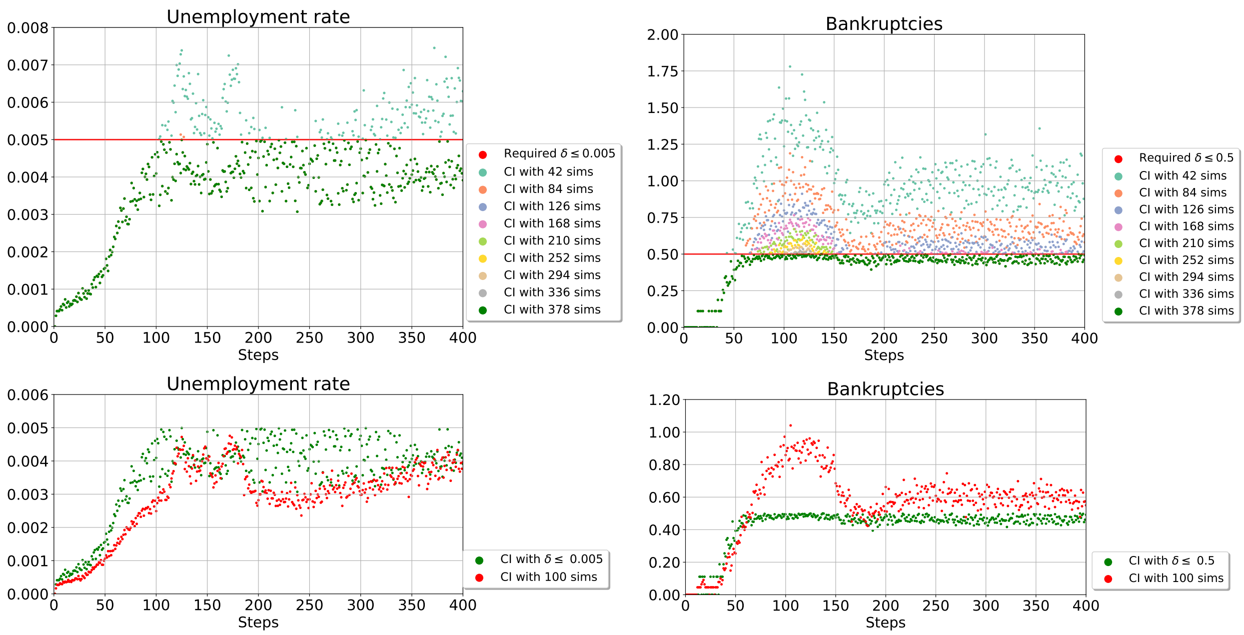


Figure 8: Top: Intermediate widths of the 97.5% CI for bankruptcy and unemployment rate computed by MultiVeStA (top); Bottom: Comparison among CI widths computed by MultiVeStA and by setting 100 simulations for all properties and *t* as in Caiani et al. (2016) (obtained using `autoIR` and setting both *bl* and the maximum number of simulations performed to 100).

and time point should be studied using its *own best* number of simulations, confirming that a trade-off between an insufficient and an excessively large number of simulations exists. When this is too low the across-runs variability might not be adequately washed-out and the representation of (stochastic) uncertainty could depend on the level of the relevant variable; conversely, when the number of simulations is too large, simulations are redundant and the same representation of uncertainty can be effectively offered saving computational time. Finally, in line with Secchi and Seri (2017), the right-hand panels of both figures confirm that the arbitrary choice of $n = 100$, which is common in the literature (see the discussion in Sections 1 and 2), is unjustified by the properties of the model itself.

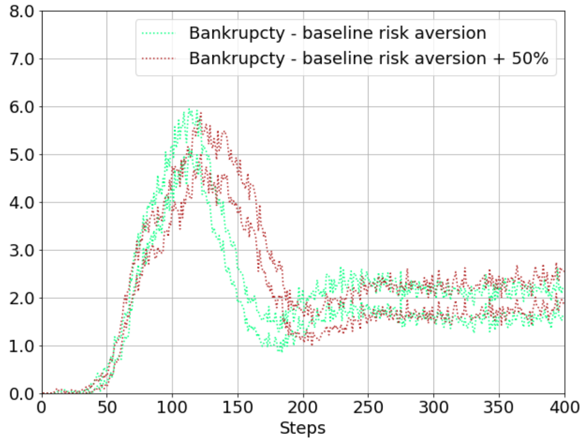
6.3. Automatic experiment comparison and statistical testing

The second exercise we perform uses the confidence intervals previously computed (and the means, variances and number of samples returned by MultiVeStA) to automatize a series of tests that identify statistical differences across model configurations discussed in Section 3.1.2. Indeed, one of the most common approaches in the macro ABM literature is to focus on key parameters or mechanisms of interest - often reflecting either behavioural attitudes or policy strength - and test how the dynamics of the model respond to changes. Just to make few examples, Dosi et al. (2015) compare a series of rules for monetary and fiscal policy, Lamperti et al. (2020) explore feed-in tariffs and R&D subsidies, and Caiani et al. (2019) extend the model analysed in this section to study various progressive tax schemes and their effects on growth and inequality. The difference across experiments is tested comparing the value of some statistic of interest (e.g. the growth rate of output) - usually averaged over the entire time span - by means of t-tests (e.g. in Dosi et al., 2015; Popoyan et al., 2020). Obviously, the ability of the test to discern across experiments and to validate the counter-factual policy intervention is affected by the choice of n, as an insufficient number or runs is likely to make model configurations (i.e. experiments) difficult to distinguish. Even further, it is not infrequent that statistical tests about differences across experiments are completely missing (e.g. Cincotti et al., 2010; Caiani et al., 2019), which weakens the potential of the paper and the eventual policy recommendations.

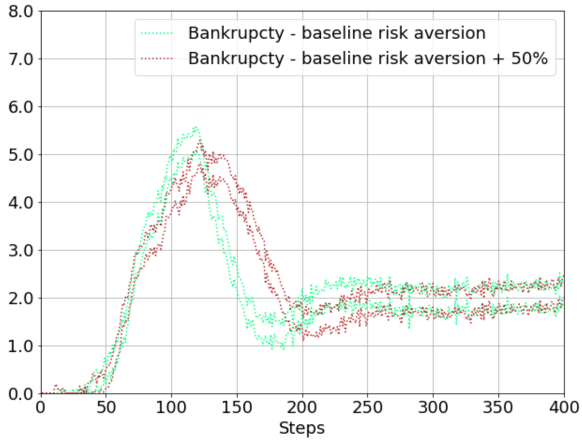
Our tool-box provides an automatic series of t-tests across experiments, where the expected value of any variable of interest in any pre-determined set of experiments is tested against a baseline configuration for each step of the transient period. As discussed in Section 3.1.2, tests are run post-mortem and consist in Welch's t-tests (Equation (2)), whose power can be computed with respect to a minimum distance ε between the means that the test is expected to detect (Equation (3)). Figure 9 shows the results in our test-bed macro ABM. Among different possible experiments, we evaluate the effects that changes in the degree of agents' risk aversion (C) produces on the number of bankruptcies and the unemployment rate. As showed by Caiani and co-authors, when risk aversion of the agents increases, the economy tends to completely avoid the recession phase experienced in the baseline configuration. While they did not offer a statistical analysis of these differences, our approach automatically embeds it. In particular, we contrast model behaviours across the baseline value of C and a 50% increase of the latter. Figure 9 shows the results of our tests comparing the set-up of the original analysis (i.e. with $n = 100$ for all properties and time points, left column) to our approach (n automatically determined for each property and time point, right column). While increasing risk aversion delays the peak of bankruptcy rates, we show that no statistical difference between the two experiments is found but for the central part of the simulation, that is when the economy first experiences a deep crisis and then recovers (see Figure 2 in Caiani et al., 2016). This is evident in both set-ups and suggests that doubling risk-aversion modifies the shape of the crisis (smoother surge of bankruptcies and slower decline) but not its existence nor duration, which is further confirmed by the behaviour of the unemployment rate (see Figure 10).[17]

Though using $n = 100$ or our approach makes little difference in terms of type 1 errors, a key advantage is evident when comparing powers. Indeed, our setup guarantees a much higher power of the tests, thereby reducing dramatically the chance of not rejecting the null hypothesis of equality across experiments when it is actually false (see also Secchi and Seri, 2017). We notice that the same holds for the unemployment rate (see Figure 10), though the discrepancy is less marked. Further, our approach delivers - for given significance a_w and setting ε equal to the δ used for the transient analysis - a *good* and stable power across the simulation horizon, i.e. above 0.8, which is usually considered an acceptable threshold in the applied statistics literature (see e.g. Secchi and Seri, 2017; Cohen, 1992; Lehr, 1992). This comes by the fact that for each property and time point MultiVeStA had to run the *correct* number of samples to obtain a constant width of the CI embedded in the choice of δ (and by the assumption that the minimum difference we want to detect – ε – is equal to δ). Indeed, it is interesting to note how the t-test for bankruptcies obtained for the setting with 100 simulations (Figure 9 bottom-left) has a low power which appears to decrease specularly to how the corresponding CI width increases in Figure 8 (bottom-right). Hence, we can derive a rule of thumb to support the modeller's choice of the two free parameters in a set-up that compares different experiments: first of all, a_w can be set equal to the α used for the transient analysis, from 5% to 1%, whose extrema are the most diffused levels of statistical significance in the social sciences. Then, by setting δ in the transient analysis equal to the ε of interest, we expect to obtain t-tests with good power. If this is not the case, one can perform the transient analysis for smaller values of δ while keeping constant ε. In case the maximum budget of simulations that have been originally chosen does not allow to meet such conditions, a trade-off exists in accepting an higher chance of type II error (not detection of false negatives) and the computational resources at disposal of the modeller. However, we stress that while increasing the size of the simulation exercise might come at the expenses of computational time, the proposed tool automatically

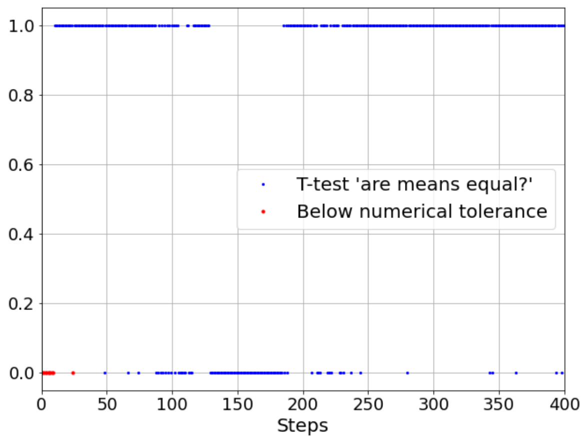
[17] We also highlight that our approach identifies a statistically significant difference between the two experiments for the slight increase in business insolvencies between period 200 and 250.



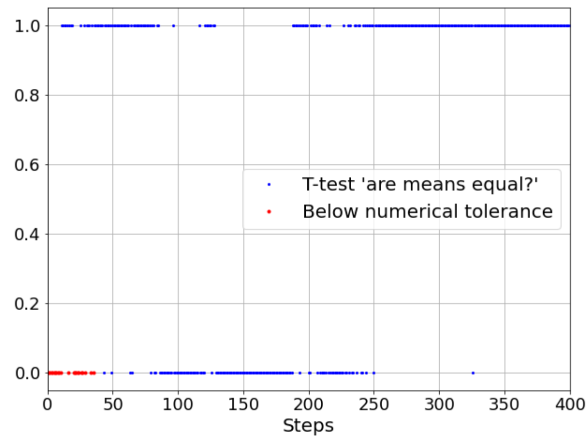
(a) CIs width for $\alpha = 0.025$ and $N = 100$ simulations



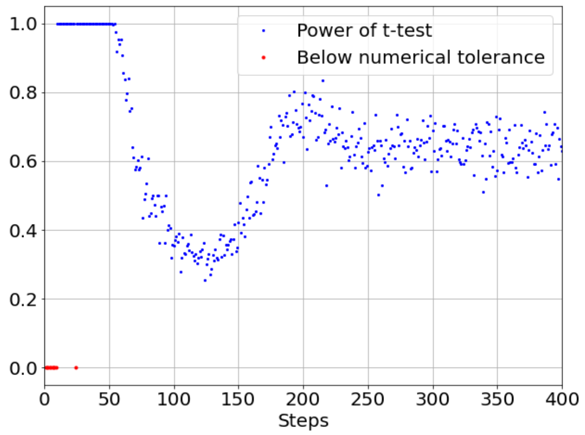
(b) CIs width for $\alpha = 0.025$ and $\delta = 0.5$



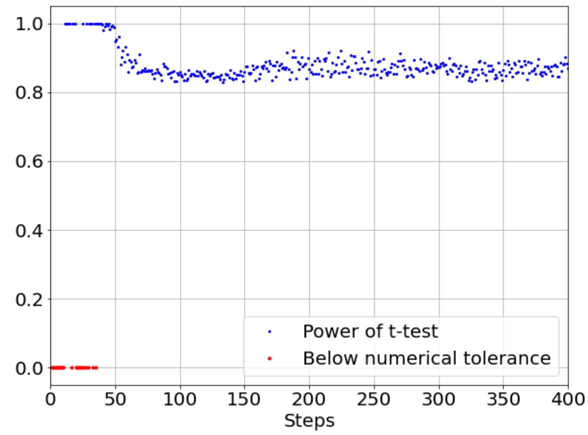
(c) T-test are means in (a) point-wise equal? - significance $a_w = 0.025$



(d) T-test are means in (b) point-wise equal? - significance $a_w = 0.025$

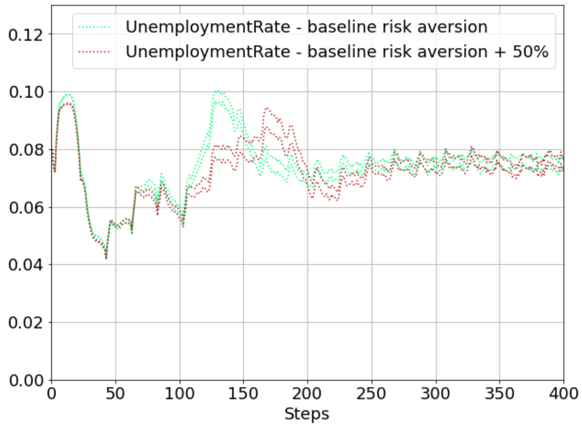


(e) Power of t-test in (c) for difference $\varepsilon = 0.5$

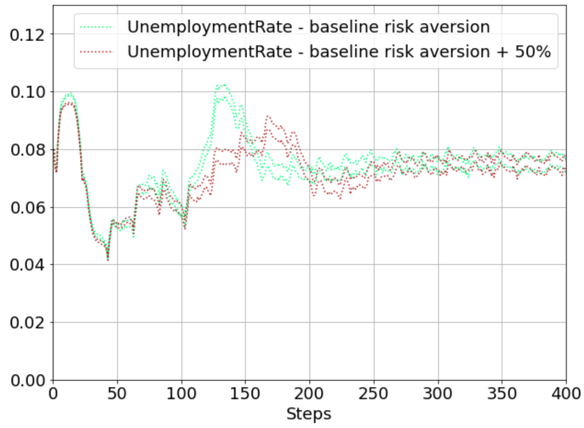


(f) Power of t-test in (d) for difference $\varepsilon = 0.5$

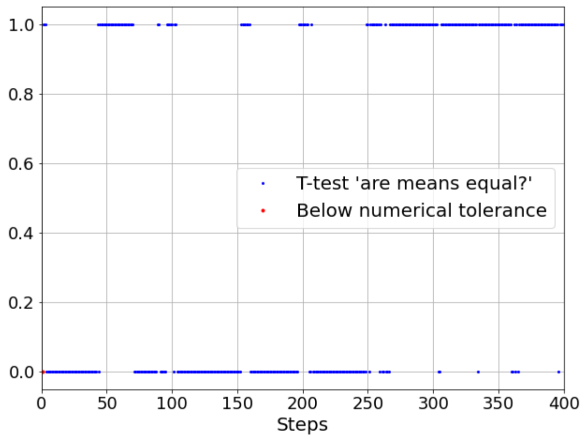
Figure 9: Evolution of bankruptcies for different risk aversions for consumption firms: are they point-wise equal? Red dots denote initial steps with variances so small to get intermediate results below the numerical tolerance of our implementation of the test (1E-15).



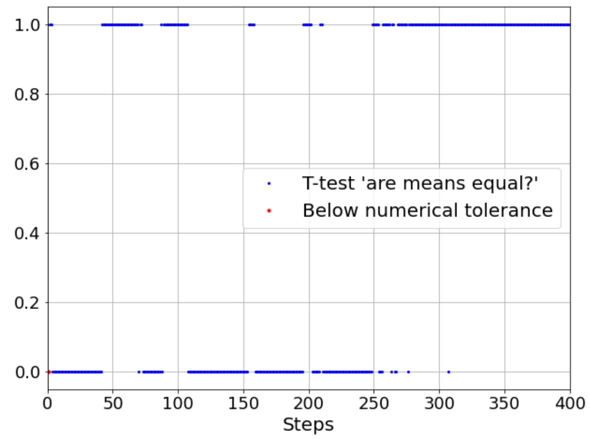
(a) CIs width for $\alpha = 0.025$ and $N = 100$ simulations



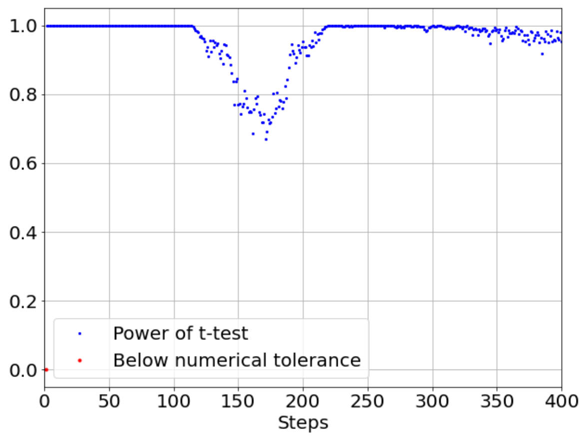
(b) CIs width for $\alpha = 0.025$ and $\delta = 0.005$



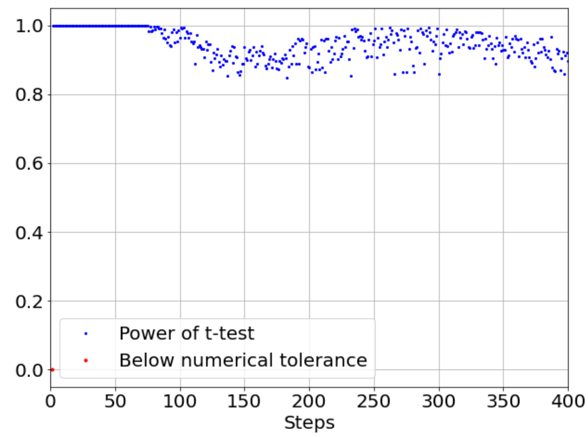
(c) T-test are means in (a) point-wise equal? - significance $a_w = 0.025$



(d) T-test are means in (b) point-wise equal? - significance $a_w = 0.025$



(e) Power of t-test in (c) for difference $\varepsilon = 0.005$



(f) Power of t-test in (d) for difference $\varepsilon = 0.005$

Figure 10: Evolution of unemployment rate for different risk aversions for consumption firms: are they point-wise equal? Red dots denote initial steps with variances so small to get intermediate results below the numerical tolerance of our implementation of the test (1E-15).

parallelise model's runs to speed-up the analysis. Section 9 shows the efficiency of our approach.

Finally, we remark that when using the original code and simulation environment (JMAB) of the Caiani et al. paper alone, it is not possible to perform any form of statistical analysis automatically, requiring to process CSV files created by the framework. Our integration of MultiVeStA provides JMAB with analysis capabilities described so far, encompassing both the transient and the steady state analysis, while leaving the simulation environment unaltered.

7. Application: Steady state analysis in a model of market selection

Here we use MultiVeStA to perform a statistical analysis of the steady state expected value of wealth shares and market price in a simple repeated prediction market model. The model we consider has been extensively studied in the literature (Beygelzimer et al., 2012; Kets et al., 2014; Bottazzi and Giachini, 2017, 2019b) and offers a perfect testbed for our procedures for automated steady state analysis. In particular, its steady state properties have been numerically investigated by Kets et al. (2014) and, later on, studied analytically in Bottazzi and Giachini (2019b) showing that the numerical results of Kets et al. (2014) were inaccurate both qualitatively and quantitatively. As briefly reported by Bottazzi and Giachini (2019b) and as we shall see, the source of inaccuracy can be traced back to the strong autocorrelation and initial condition bias that process possesses. The *post-mortem* nature of the numerical analysis carried on by Kets et al. (2014) is unable to properly deal with those issues. Our approach, instead, uses statistical tests and procedures able to manage both autocorrelation and the initial condition bias in an automated way. Thus, in what follows, we first introduce the model, then we repeat the numerical analyses of Kets et al. (2014) showing how and why the inaccuracies emerge, and finally we use `autoRD` and `autoBM` to accurately perform the steady state analyses. In fact, we match the correct analytical results from Bottazzi and Giachini (2019b).

7.1. The prediction market model by Kets et al. (2014)

The model consists in a pure exchange economy in discrete time, indexed by $t \in \mathbb{N}$, where N agents repeatedly bet on the occurrence of a binary event. That is, in every t two contracts are available for wagering: the first pays 1 dollar if the event occurs and zero otherwise, while the second pays 1 dollar if the event does not occur and zero otherwise. We model the event by means of a Bernoulli random variable s_t, such that $s_t = 1$ means that the event at time t has occurred, and $s_t = 0$ otherwise. The probability of observing $s_t = 1$ is a constant $\pi^* \in (0, 1)$. Every agent $i \in \{1, 2, \dots, N\}$ assigns a subjective probability π^i to the realization of the event at any time t. Agent i has initial wealth equal to w_0^i and at the end of every betting round it evolves in w_t^i depending on the results of her betting. The total initial wealth in the market is normalized to 1, such that, since wealth is only redistributed by the betting system, it is $\sum_{i=1}^N w_t^i = 1$ for all t.[18] Every agent i bets on the occurrence of the event at time t a fraction α_t^i of her wealth w_{t-1}^i, while $1 - \alpha_t^i$ is the fraction bet against the occurrence. As in Kets et al. (2014); Bottazzi and Giachini (2017, 2019b), we focus on the so-called fractional Kelly rule, that is $\forall i, t$

$$\alpha_t^i = c\pi^i + (1 - c)p_t\,, \tag{6}$$

with $c \in (0, 1]$. In every period, the agents exchange contracts in the competitive market, thus contracts' prices are fixed by means of market clearing conditions. Without loss of generality, we assume that contracts are in unitary

[18]Hence, one can indifferently refer to w_t^i as both the wealth and the wealth share of agent i at time t.

		Beliefs			*Wealth*		
N	c	π^1	π^2	π^3	w_0^1	w_0^2	w_0^3
3	0.01	0.3	0.5	0.8	0.33	0.33	0.34

Table 1: Parameters used for the prediction market model.

supply. Hence, calling $p_{1,t}$ and $p_{2,t}$ the price of the first and second contract, respectively, we have $\forall t$

$$1 = \sum_{i=1}^{N} \frac{\alpha_t^i}{p_{1,t}} w_{t-1}^i \quad \text{and} \quad 1 = \sum_{i=1}^{N} \frac{1-\alpha_t^i}{p_{2,t}} w_{t-1}^i .$$

Since wealth sums up to 1 in every period, one has $p_{1,t} + p_{2,t} = 1$, hence we call $p_{1,t} = p_t$ and $p_{2,t} = 1 - p_t$. Substituting with Equation (6) and applying simple algebraic manipulations, one obtains

$$p_t = \sum_{i=1}^{N} \pi^i w_{t-1}^i \quad \forall t . \tag{7}$$

After the market round, the outcome of the binary event is revealed and the wealth of agent i evolves according to

$$w_t^i = \begin{cases} \dfrac{\alpha_t^i}{p_t} w_{t-1}^i = \left(1 - c + c\,\dfrac{\pi^i}{p_t}\right) w_{t-1}^i & \text{if } s_t = 1 , \\ \dfrac{1-\alpha_t^i}{1-p_t} w_{t-1}^i = \left(1 - c + c\,\dfrac{1-\pi^i}{1-p_t}\right) w_{t-1}^i & \text{if } s_t = 0 . \end{cases} \tag{8}$$

In this setting, Kets et al. (2014)[19] want to explore the selection dynamics of the model and are particularly interested in the asymptotic (steady state) value of expected wealth shares and price for $c \to 0$. Indeed, they conjecture that in such a limit the steady state expectation of p_t matches π^* and use the case $c = 0.01$ as a proxy.

In Section 7.2, we replicate exactly the analysis of Kets et al. (2014), reproducing their Figures 3(c) and 3(d). Thus, we follow the procedure proposed in Kets et al. (2014) to estimate steady state expected wealth shares and price for several values of π^* under the parametrization in Table 1. In doing that, we highlight some issues related to initial condition bias and strong autocorrelation. Indeed, the warmup period appears not correctly determined, and the autocorrelation within the observations of each performed simulation not correctly accounted for. This is due to the procedure for computing CIs used in Kets et al. (2014) and has the result of producing extremely wide CIs.

After this, in Section 7.3, we perform the steady state analyses using our approach. We show that, thanks to the provided automatic procedures, our estimates (and the corresponding confidence intervals) of steady state expected wealth shares and price are correctly determined. Our conclusions differ not just quantitatively, but also qualitatively from those in Kets et al. (2014), and match those from Bottazzi and Giachini (2019b). Overall, our analysis shows the importance of using an automated procedure provided with statistical guarantees.

7.2. Steady state analysis with `manualRD` *using original wrong warmup estimation*

As discussed in Section 5.3, by using `manualRD` MultiVeStA allows one to manually set an *a-priori* estimate of the warm-up period. MultiVeStA also allows one to fix the maximum number of simulations used in an analysis based on

[19]Notice that the mathematical specification of the model may appear different from the one presented in Kets et al. (2014). However, as explained in Bottazzi and Giachini (2019b), the two specifications are indeed equivalent.

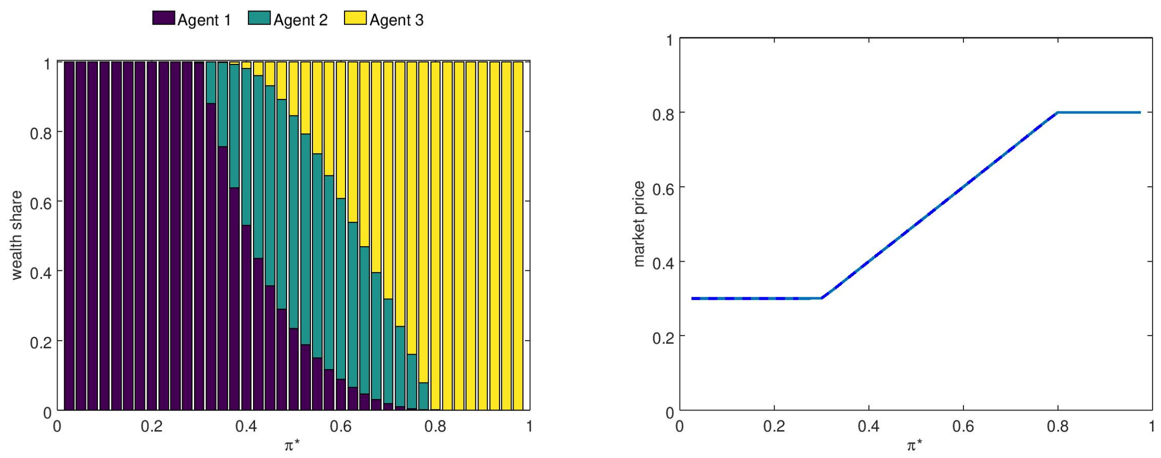


Figure 11: Steady state analysis of expected agents' wealth and market price according to the manual warm-up and simulation length settings of Kets et al. (2014). We obtain these results by using `manualRD` setting the end of the warmup periods to 90 000, and the time horizons to 100 000. By setting both *bl* (the number of simulations in a block) and the maximum number of simulations performed to 1 000, we use precisely 1 000 simulations to estimate each property, perfectly matching the setting used in Kets et al. (2014). We consider 39 equally spaced values for π^*, from 0.025 to 0.975, each requiring a separate MultiVeStA analysis on a correspondingly parameterized instance of the model (we automated this process using an external Octave script). Confidence intervals computed by MultiVeStA for agents' wealth, not reported in the left panel, are such that the maximum recorded width for a statistical confidence of 90% is below 0.0025. Confidence intervals for the market price are reported in the right panel, with maximum recorded width below 0.00065 for statistical confidence of 90%.

independent replication. In general it is always advisable to do not fix such parameters *a-priori*, but to use the offered automated procedures so to avoid bias in the estimates and excessively large CIs. In this section we exemplify these issues by fixing a priori the erroneous warmup estimate and number of simulations used in Kets et al. (2014), and discuss the problems this introduced in the obtained results.

Kets et al. (2014) performed an RD-based steady state analysis of the agents' wealth (w_t^1, w_t^2, w_t^3) and market price p_t using the parametrization of Table 1. The authors arbitrarily estimated the end of the warmup period after 90 000 steps, and fixed the time horizon of each simulation to 100 000 steps and the number of performed simulations to 1 000. This means that estimates were computed averaging the last 10 000 observations in each simulation (the horizontal means of Figure 1(c)) and then further averaging the so-computed means from each simulation (the vertical means). We shall see how this led to estimates highly biased by the initial conditions.

Regarding computations of confidence intervals, Kets et al. (2014) did not follow the standard approach relying on the central limit theorem used by MultiVeStA. Rather, Kets et al. (2014) considered how the above discussed 1 000 averages built for each simulation distribute. In particular, the 5-th and 95-th percentiles of such distribution are taken as the bounds of confidence intervals with 10% statistical significance. The problem with this approach is that, differently from the approach used by MultiVeStA, it is based on the assumption that each of the considered 1 000 averages has the same distribution of an average across independent replications computed at a time t large enough to have reached steady state. This is not correct because, as well as the initial condition bias, the process is characterized by strong autocorrelation. We shall discuss how this led to erroneous interpretation of the results.

Agents' wealth. In Figure 11 we report the outcomes of the exercise replicating those from Figures 3(c) and 3(d) in Kets et al. (2014) considering model variants for 39 different values of π^*. Looking at the left panel one should conclude that there exist model configurations in which all agents have strictly positive expected wealth share in steady

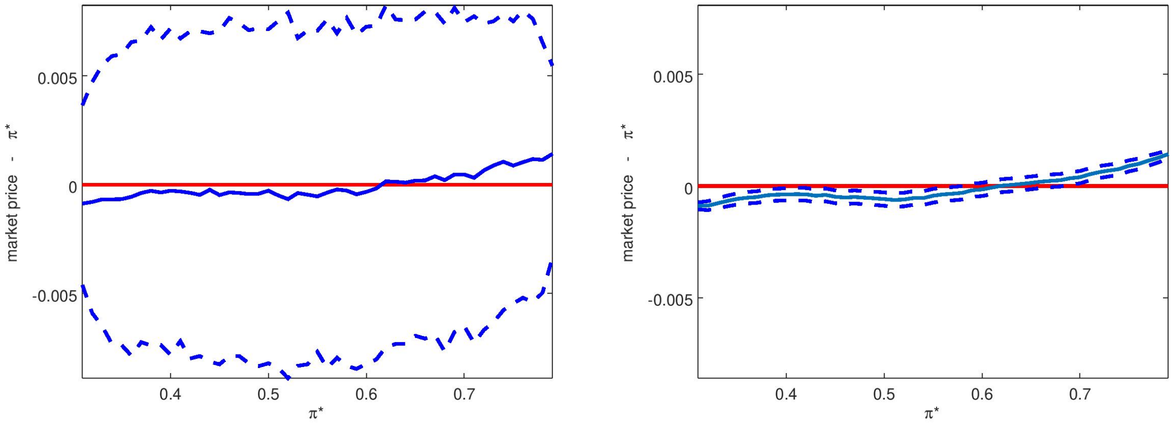


Figure 12: Estimates of steady state difference between the expected price and π^* using the settings from Figure 11. We consider 49 equally spaced values for π^*, from 0.31 to 0.79. The dashed lines are CIs built on the same samples using two different procedures. Left: CIs are erroneously computed (using an external Octave script) according to the procedure in Kets et al. (2014). Right: CIs as computed by MultiVeStA using the approach based on the central limit theorem.

state. This is, however, in contrast with the analytical analysis from Proposition 4.1 of Bottazzi and Giachini (2019b), which proves that no more than two agents can have asymptotic positive wealth share. Thus, the fact that Kets et al. (2014) incorrectly suggest that more than two traders can have positive expected wealth in steady state is an artifact of the initial condition bias that affects their analysis. Notice that the convergence to zero of the wealth share of at least one trader is asymptotic, thus wealth shares show a bias for any t. However, such bias decreases with t and can be made negligible choosing a sufficiently long warm-up and simulation length. What we observe is that discarding the first 90 000 observation of every run is simply not enough.

Market price. The right panel of Figure 11 shows the average price and should support one of the main results of Kets et al. (2014): the expected market price matches π^* when $c = 0.01$ and π^* is strictly between the lowest and the highest of agents' beliefs. Bottazzi and Giachini (2019b) suggest that such a conclusion is not correct and the source of inaccuracy should be found in the way in which CIs are built by Kets et al. (2014). In order to better understand this aspect, we create a new plot in Figure 12 focusing on the difference between market price and π^*. In the left panel of the figure we report the CIs (dashed lines) obtained by applying the procedure of Kets et al. (2014), while in the right panel we show those obtained by MultiVeStA. As one can notice, the procedure of Kets et al. (2014) produces large CIs, backing the claim of the authors. Instead, the CIs obtained by MultiVeStA using the approach based on the central limit theorem are much tighter and disprove the claim of the authors. To understand the source of disagreement between the two approaches, consider the following argument: if the 10 000 observations of p_t from each simulation used to compute the average price of every replication were independent and at steady state, then we would not have spotted any significant difference. Indeed, according to the Central Limit Theorem, we have that each time average is (approximately) distributed as a normal random variable with mean the steady state expectation of the price and variance the steady state variance of price over $\sqrt{10\,000}$. The initial condition bias lets the expected time average be different from the steady state expectation. The strong autocorrelation in the price process (Bottazzi and Giachini, 2019b) lets confidence intervals be too wide. While the `manualRD` procedure used here can do nothing about the initial condition bias, with respect to confidence intervals MultiVeStA does not assume anything about the distribution of time averages, simply relies on the Central Limit Theorem. Indeed, the average across 1 000 independent replications

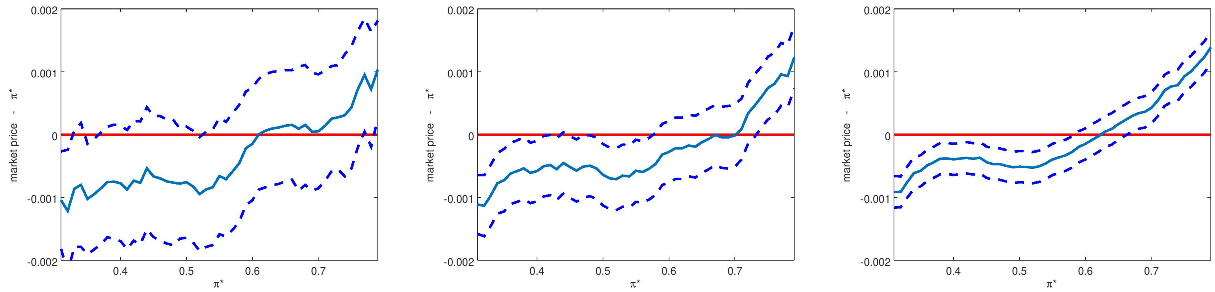


Figure 13: Estimates of steady state difference between the expected price and π^* using the settings from Figure 11 for warmup estimation and time horizon. The number of simulations is automatically chosen by MultiVeStA according to the used values of δ, for $\alpha = 0.025$. As in Figure 12, we consider 49 equally spaced values for π^*, from 0.31 to 0.79. Left: $\delta = 0.002$, the number of simulations varies between 60 and 120. Center: $\delta = 0.001$, the number of simulations varies between 120 and 360. Right: $\delta = 0.0005$, the number of simulations varies between 420 and 1 440.

of the 10 000-period time averages is (approximately) distributed as a normal random variable with variance the time averages' variance over $\sqrt{1\,000}$.

This exercise shows the importance of correctly building confidence intervals when testing hypothesis on steady state quantities from simulated models. We proceed showing that setting the required statistical significance (α) and confidence interval width (δ) instead of the total number of independent replicas is a much more reliable and efficient procedure to test hypotheses on steady state expectations.

Market price for different α-δ. In Figure 13, we test the hypothesis from Kets et al. (2014) that no difference between the average price and π^* exists under the parametrization in Table 1. We use again `manualRD` keeping the same settings for warmup estimation and time horizon discussed in advance, while the number of simulations is automatically chosen by MultiVeStA according to different values of δ (0.002, 0.001, and 0.0005), for $\alpha = 0.025$ (i.e. a statistical confidence of 97.5%). If the hypothesis from Kets et al. (2014) were correct, the difference should be almost never significantly different from zero for any δ considered. Instead, we notice that δ plays an important role in assessing the hypothesis testing outcome. Indeed, while with $\delta = 0.002$ the computed CIs for the difference among market price includes 0 for almost all π^*, with $\delta = 0.001$ the CIs almost never includes 0. This confirms the point of Bottazzi and Giachini (2019b) and the results we have obtained in Figure 12 right panel: the hypothesis of no difference between the average price and π^* is generically rejected. Focusing on $\delta = 0.0005$ and looking at the number of required simulations, one notices that there exist cases in which the hypothesis that no difference between the average price and π^* exist can be rejected with less than 1 000 simulations[20]. In other cases, instead, 1 000 independent replications are not enough and one may risk to get to the wrong conclusion simply because of an insufficient number of replicas.[21]

Notice, however, that due to the arbitrary choice of the end of the warmup period, all the estimates are biased by the initial conditions. We next use MultiVeStA's automated steady state analysis (`autoRD` and `autoBM`) to accurately estimate steady state expectations and to finally assess on such obtained results the hypothesis of Kets et al. (2014).

7.3. Steady state analysis with `autoRD` *and* `autoBM` *using automatic warmup estimation*

Now we repeat the exercises using the automated tools for steady state analysis provided by MultiVeStA, starting by estimating expected wealth shares.

[20] This is typically the case at the extrema, indeed 0.31 and 0.79 require, respectively, 420 and 480 replicas.
[21] This may occur for π^* around 0.55, where MultiVeStA needs 1 440 simulations to reach the required interval width.

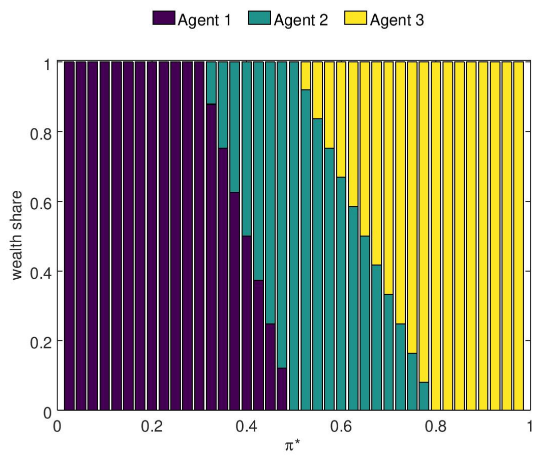
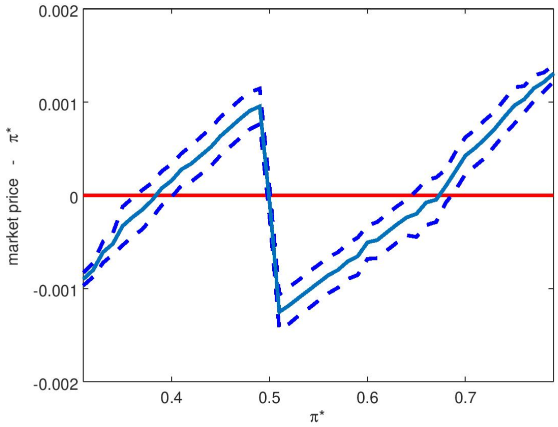


Figure 14: Steady state levels of average wealth shares. Left: Replication-Deletion. Right: Batch Means. We set $\alpha = 0.025$ and $\delta = 0.001$, while we consider 39 equally spaced values for π^*, from 0.025 to 0.975, each one of these points requires a separate MultiVeStA query that has been invoked and aggregated with the others by means of an external Octave script.

Agents' wealth. Our results are displayed in Figure 14. In the left panel we use the Replication-Deletion approach (`autoRD`) while on the right we use the Batch Means approach (`autoBM`). As one can notice: *i*) our results comply with the theoretical and numerical ones of (Bottazzi and Giachini, 2019b, cf. Figure 7) and *ii*) no significant difference can be spotted between the two pictures. Hence, our automated procedures allow one to avoid (or, at least, reduce) biases generated by initial conditions. As a practical example, let us consider the statistical analysis of steady state expectations for $\pi^* = 0.6$. The `manualRD` procedure, using the settings from Section 7.2, estimates the expected wealth share of agents 1, 2, and 3 to, respectively, 0.089, 0.519, and 0.392. Instead, `autoRD` and `autoBM` estimate the expected wealth shares to, respectively, 0, 0.668, and 0.332, in agreement with the results from Bottazzi and Giachini (2019b). Looking at the estimated warm-up end, our automated tools propose values higher than 4 000 000 for every expected wealth share. This confirms how manually setting the warm-up end to 90 000 generates a large initial condition bias in the estimation of steady state expectations of agents' wealth. Our analysis, other than correctly estimating steady state expected wealth shares, clearly highlights the source of inaccuracy in the exercise of Kets et al. (2014), and stresses the importance of using a reliable automated procedure to pursue steady state analyses.

While there are no significant differences between the estimates generated by `autoRD` and autoBM, the time required for producing the results changes. Indeed, the analysis runtime of `autoRD` (using parallelism degree 3) is about 3 times larger than the one of `autoBM`. As discussed in Section 3.2.4, cases like this with large warmup periods tend to favour `autoBM`.

Market price. Next, we estimate the expected value in steady state of the market price. As we did in Figures 12 and 13, in order to magnify confidence bands in Figure 15 we show our estimates of the difference between the expected price and π^* for different values of $\pi^* \in (0.3, 0.8)$. In the left and right panels we show the `autoRD` and `autoBM` results, respectively. We notice that, for any value of π^* considered, the estimated difference between the expected price and π^* does not change in a significant manner between the two plots. The emerging expected difference presents a clear pattern: it is larger (in absolute value) when π^* is close to the belief of one of the agents. Moreover, the expected difference appears to be negative when π^* is closer to the belief of the agent whose belief is, relatively, the smallest (i.e. among the surviving ones) while it tends to be positive when it is the other way round. These features are in



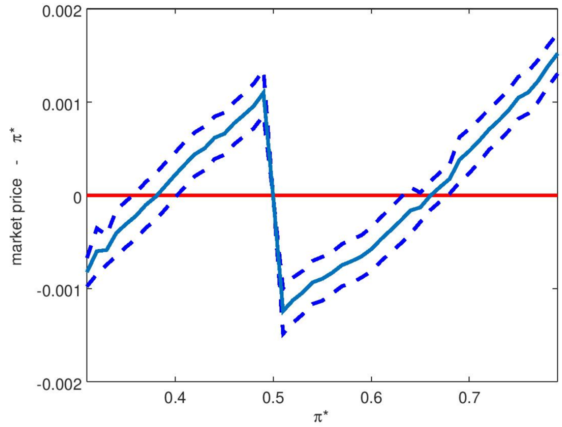


Figure 15: Steady state levels of average price. Left: Replication-Deletion. Right: Batch Means. We set $\alpha = 0.025$ and $\delta = 0.0005$, while we consider 49 equally spaced values for π^*, from 0.31 to 0.79, each one of these points requires a separate MultiVeStA query that has been invoked and aggregated with the others by means of an external Octave script.

line with the results obtained by Bottazzi and Giachini (2019b) in Figure 4. Hence, the reliability of the steady state analysis performed by MultiVeStA is confirmed. Moreover, we can conclude that, contrary to what Kets et al. (2014) argue, the steady state value of the average price does not generally match π^* when $c = 0.01$.

The analysis presented in Section 7.3 satisfies all tests of the methodology for ergodicity diagnosis from Section 4, confirming the reliability of the analyses. We show in the next section examples of analysis where this does not hold.

8. Application: Ergodicity diagnosis in a CRRA prediction market model with noise

We apply our methodology for ergodicity diagnosis to variants of the prediction market model. For all analyses we set $\alpha = 0.05$ and $\delta = 0.01$.

8.1. Three variants of the prediction market with 2 CRRA traders: IID noise, AR noise, ergodic

Here we modify the model studied in the previous section to allow violations of the ergodicity assumption. Following Bottazzi and Giachini (2019a), it is enough to assume that in the market there are $N = 2$ traders who bet maximizing their next-period CRRA utility to obtain non-ergodic price and wealth dynamics. Such a different behavioral assumption changes the betting rules. Indeed, we keep the assumption that agents 1 and 2 have heterogeneous beliefs (π^1 and π^2, respectively, with $\pi^1 < \pi^2$) and we add risk preferences, assuming that the relative risk aversion coefficient of agent i is $\gamma^i > 0$, with $i = 1, 2$. Thus, we replace eq. (6) with

$$\alpha_t^1 = (1 - b_t^1)p_t \quad \text{and} \quad \alpha_t^2 = (1 - b_t^2)p_t + b_t^2\,, \quad \text{where} \tag{9}$$

$$b_t^1 = \frac{\left(p_t(1-\pi^1)\right)^{\frac{1}{\gamma^1}} - \left(\pi^1(1-p_t)\right)^{\frac{1}{\gamma^1}}}{\left(p_t(1-\pi^1)\right)^{\frac{1}{\gamma^1}} + p_t\left(\pi^1\right)^{\frac{1}{\gamma^1}}(1-p_t)^{\frac{1-\gamma^1}{\gamma^1}}} \quad \text{and} \quad b_t^2 = \frac{\left(\pi^2(1-p_t)\right)^{\frac{1}{\gamma^2}} - \left(p_t(1-\pi^2)\right)^{\frac{1}{\gamma^2}}}{\left(\pi^2(1-p_t)\right)^{\frac{1}{\gamma^2}} + (1-p_t)\left(1-\pi^2\right)^{\frac{1}{\gamma^2}}(p_t)^{\frac{1-\gamma^2}{\gamma^2}}}\,. \tag{10}$$

Bottazzi and Giachini (2019a) show that, depending on the parameters values – in particular γ^1 and γ^2 – several long-run selection scenarios are possible. Indeed, one can generically have that: *i*) one of the two agent has asymptotic

Scenario			*Beliefs*		*Risk Aversion*		*Wealth*		*Noise*	
	N	π^*	π^1	π^2	γ^1	γ^2	w_0^1	w_0^2	η	θ
IID noise	2	0.45	0.2	0.5	2	0.5	0.5	0.5	0.5	0
AR noise	2	0.45	0.2	0.5	2	0.5	0.5	0.5	0.5	0.9
Ergodic	2	0.45	0.2	0.8	2	2	0.5	0.5	0.5	0.9

Table 2: Parameters used for the prediction market model with CRRA traders and noisy price reporting.

unitary wealth share, *ii*) both agents maintain positive wealth share asymptotically, *iii*) path dependent scenarios in which either agent 1 obtains unitary wealth share asymptotically while agent 2 loses everything or vice-versa. Focusing on the market price p_t, in case *i*) p_t converges to π^i (with i the dominating agent), in case *ii*) p_t fluctuates in the interval (π^1, π^2), and in case *iii*) p_t either converges to π^1 or to π^2 depending on the particular sequence of events realized. Case *iii*) is the one we are interested in: in such a case the ergodicity assumption is violated. However, the asymptotic convergence of the price to one out of two points makes quite easy to spot the lack of ergodicity and the presence of the two possible long-run price values. Hence, we complicate the setting assuming that there exists a third agent in the model who does not trade nor interacts in any way with agents 1 and 2. He simply observes the price and reports it. Such report is, however, noisy. Defining $\tilde{p}_t$ the price such external agent reports, we assume $\tilde{p}_t = p_t + v_t$ with $v_t = \theta v_{t-1} + u_t$ and u_t a uniformly distributed random variable: $u_t \sim \mathcal{U}(-\eta, \eta)$, $\eta > 0$. Such price reports are not taken into account by agents 1 and 2, hence all the properties of p_t deriving from the analysis of Bottazzi and Giachini (2019a) remain unaffected. Moreover, v_t is an autoregressive process of order 1 with zero mean. Hence, assuming $|\theta| < 1$, we have that in the long-run $\tilde{p}_t$ fluctuates around either π^1 or π^2 depending on the sequence of realized events. Thus, the lack of ergodicity $\tilde{p}_t$ shows is somehow "well-behaved". That is, if one isolates the sequences in which p_t converges to a given π^i, one will obtain that the time averages (i.e. horizontal means) of the relative observations of $\tilde{p}_t$, for t large enough, are approximately normally distributed with mean π^i. Hence, we can say that $\tilde{p}_t$ presents two stationary points. At the same time, studying ergodicity of $\tilde{p}_t$ is much more complicated than performing the same tasks on p_t. In what follows, we set $\eta = 0.5$ and consider two scenarios for θ. In the first one, we consider $\theta = 0$. We refer to it as "IID noise" scenario, since we have that, in the long-run, the fluctuation described by $\tilde{p}_t$ around either π^1 or π^2 are IID. In the second scenario, instead, we consider the opposite case: setting $\theta = 0.9$ we analyze the performance of our methodology when the noise is highly autocorrelated. We refer to it as the "AR noise" scenario. Finally, as a robustness check, we apply our methodology to a case in which ergodicity should be ensured. We choose a scenario belonging to case *ii*): long-run survival of both agents. This makes p_t fluctuate in the interval (π^1, π^2) indefinitely. Moreover, we set $\theta = 0.9$ as in AR noise. These two assumptions, even if not affecting the ergodic properties of $\tilde{p}_t$, should make relatively harder for our methodology to work. We refer to this case as "Ergodic". Table 2 summarizes the parametrization used in our analyses. While the setting for *IID noise* and *AR noise* scenarios ensure the emergence of multiple stationary points, leading to a non-ergodic scenario, the assumptions for the *Ergodic* scenario guarantee the persistent fluctuation of p_t (Bottazzi and Giachini, 2019a).

8.2. Application of the methodology for ergodicity analysis

IID noise. We start our analysis by applying `autoBM` and `autoRD` to the IID noise case. The former requires 33 792 steps of simulation. It signals that the warmup ends after the first batch of 1 024 steps, and estimates the steady state mean as 0.498. Instead, `autoRD` signals that the warmup ends after 1 032 steps. After 2 604 independent replications, it estimates the steady state mean as 0.426. With reference to our methodology for ergodicity analysis in Figure 4,

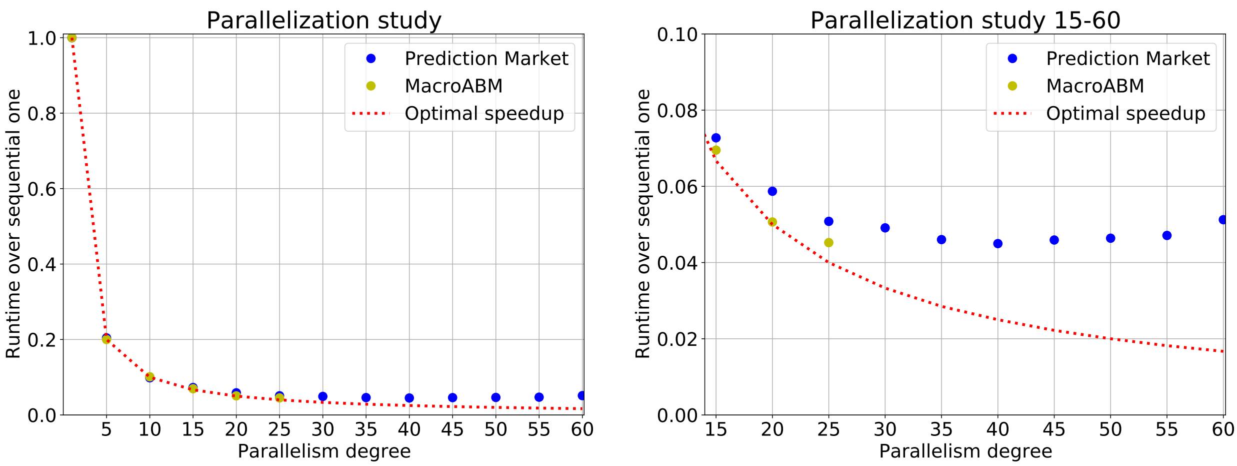


Figure 16: Analysis of the runtime speed-ups on a machine with 20 physical cores.

we performed step 1, and passed the termination check of step 2. After that, step 4 requires to compare the results of `autoBM` and `autoRD`. The difference among the two results is larger than δ, suggesting an ergodicity problem. According to our method, we already have an indication of non-ergodicity. However, for illustrative reasons we also performed the Anderson-Darling normality test on the horizontal means computed by `autoRD` (step 5), obtaining a p-value equal to 6.092E-251, which allows us to reject the null hypothesis that the horizontal means are normally distributed. Hence, our methodology is able to correctly spot that the IID noisy price lacks ergodicity.

AR noise. We consider now the AR noise case starting with `autoBM`. The algorithm estimates the warmup to end in 1 024 steps, and the steady state mean as 0.499. Instead, by performing `autoRD` we obtain that the warmup is estimated to end after 1 032 steps. The total number of independent replications needed by `autoRD` to reach the IC width is 2 709, obtaining as result 0.426. Therefore, the two algorithms provide significantly different results for the used δ, suggesting an ergodicity problem (step 4). This is confirmed by the Anderson-Darling normality test of step 5 which computes a p-value of 1.458E-136, rejecting the normality assumption. Therefore, our methodology is able to correctly spot that also the AR noisy price lacks ergodicity.

Ergodic. Finally, we perform a robustness check on our methodology by applying it to the Ergodic scenario. Using `autoBM`, the warmup is estimated to end after 1 024 steps producing as result 0.4027. Using `autoRD`, one gets that the warmup is estimated to end after 1 032 steps. The number of independent replications needed for reaching CIs of width δ is 210, obtaining as result 0.4035. Thus, the two algorithms give results within the tolerance of δ (step 4). The normality test from step 5 cannot reject the null hypothesis of normality, as we get a p-value of 0.691. Therefore, our methodology correctly suggests that no violation of ergodicity is observed.

9. Parallelization study

One of the key issues in the analysis of ABMs in social sciences concerns computational time; while some approaches have recently proposed to take advantage of machine learning surrogates (Lamperti et al., 2018; van der Hoog, 2019), the most direct approach to speed-up simulation is an efficient parallelisation of the experiments. In this section we discuss how MultiVeStA can efficiently and automatically parallelize the various runs. Notably, we demonstrate the potential analysis speedups showing an analysis that requires about 15 days when performed in sequential, and about 16 hours when parallelizing it on a machine with 20 cores. In particular, we show the actual runtime gains obtained on the analysis of our case studies when using different degrees of parallelism on a machine with 1 CPU Intel Xeon Gold 6252 (20 physical cores) and 94GB of RAM. This machine allows to perform up to 20 processes in parallel, but *hyperthreading* further allows for limited speedups also with parallelism degrees higher than 20.

Figure 16 (left) shows the results of our study considering the sequential case and parallelism degrees N multiple of 5 up to 60. Intuitively, in the ideal case an analysis using parallelism degree N should take $\frac{1}{N}$ of the time required by a sequential analysis (i.e. with $N = 1$). For this reason, the red dashed line provides the optimal obtainable speed-ups: 1 (no speed up) for the sequential case, and $\frac{1}{N}$ for all considered N. The blue and yellow dots, instead, show the actual speedups obtained for our two case studies. In particular, in order to compare with the optimal speedup, for each value of N we provide the ratio among the runtime obtained with parallelism degree N over the one of the sequential case. For the prediction market model, we consider the `autoRD` analysis from Figure 14 (left) for $\pi^* = 0.45$, while for the macro model we consider the analysis from Figure 7. Notably, the analysis of the macro model took about 15 days when executed sequentially, while it goes down to about 18 hours for $N = 20$, and 16 hours for $N = 25$. The analysis failed for higher values of N due to the high memory requirements of the model. Instead, the analysis of the prediction market model requires about 14 minutes in sequential and about 50 seconds for $N = 20$. The analysis could be performed for all considered N, with a minimum runtime of about 38 seconds for $N = 40$.

Overall, for both case studies we note speedups very close to the optimal ones up to $N = 20$, while they tend to deteriorate for higher values of N. Figure 16 (right) focuses on the values of N from 15 onwards. We see that the speedups obtained for the macro model tend to be closer to the optimal ones. This is because simulations are computationally intensive, taking more than 1 hour. Therefore, the *overhead* (i.e. the extra computations) introduced by the communications among the MultiVeStA client and servers has almost no impact on the overall runtime. Instead, the prediction market model is not particularly computationally expensive, making the extra communications influence more the overall runtime. In particular, the figure shows that relatively limited speedups are obtained for N greater than 25. This is expected, as discussed. Interestingly, increasing N further than 40 actually worsens the performances, as the processor is not anyway able to perform more than 20 processes in parallel while the overhead costs increase.

10. Conclusion

In this article we presented a fully automated framework for the statistical analysis of simulation models and, in particular, agent-based models (ABM). The framework, implemented through the statistical analyzer MultiVeStA, provides a novel toolkit to the ABM community. These tools range from transient analysis, with statistical tests to compare results for different model configurations, to warmup estimation and the exploration of steady state properties, including a procedure for diagnosing ergodicity – and hence the reliability of any steady state analysis.

Our approach can be easily applied to simulators written in Java, Python, R or C++, and we also added native support in JMAB, a framework for building macro stock-flow consistent ABMs. Our tools allow modellers to automate their explorations, save time and avoid mistakes originating from semi-automated and error-prone tasks. Importantly, this facilitates reproducibility of experiments and promotes the use of a minimal set of *default* analyses that should be performed when proposing or studying a model.

We validated our approach on two models from the literature: a large scale macro financial ABM and a small scale prediction market ABM (and variants thereof). We obtained new insights on these models, identifying and fixing erroneous results from prior analyses. Our framework also allows one to easily parallelize simulations within the cores of a machine or in a computer network. For instance, we reduced the analysis runtime for the macro ABM from 15 days to 16 hours on a machine with a CPU with 20 cores. Indeed, our toolkit enables modellers to run extensive tests in a unique environment (i.e. without the need of exporting data) and optimizing computational time (which is often precious; see also the discussion in Lamperti et al., 2018).

Our approach is rooted in results from the simulation, computer science and operations research communities, which we aim to make available to the ABM community. Connecting these communities is critical to leverage the most effective techniques and approaches across fields. For example, the stationarity analysis proposed by Grazzini (2012) mentioned in Section 2 can be viewed as a non-automated version of the batch means approach by Conway (1963) and Law and Carson (1979).

In the near future, we plan to integrate MultiVeStA with other popular platforms used to build and analyse simulation models – including the LSD environment for ABMs (Valente, 2008) and the JASMINE environment for discrete-event simulations (Richiardi and Richardson, 2017). We see this article as a first step in bringing practices from the statistical model checking (SMC) tool-set to the ABM computational economics community. Of particular interest in this respect are SMC techniques developed to mitigate two classic problems of Monte Carlo methods: dealing with models that present rare events (Legay et al., 2016), and using machine learning techniques to reduce the number of simulations (Bortolussi et al., 2015). Finally, we will expand the family of automated analysis techniques offered in MultiVeStA. For instance, we will extend and refine our ergodicity diagnostics procedure, e.g. tackling the problem of identifying multiple stationary points (assuming they are finitely many) by means of clustering algorithms. We also plan to further improve our proposals for the analysis of simulation output, e.g., by introducing corrections for multiple testing across the time domain, and move beyond it, e.g., by considering sensitivity analysis and parameter calibration, which are prominent in the ABM community.

References

Agha, G. and K. Palmskog (2018). A survey of statistical model checking. *ACM Trans. Model. Comp. Simul. 28*(1), 6:1–6:39.

Alexopoulos, C. and D. Goldsman (2004). To batch or not to batch? *ACM Transactions on Modeling and Computer Simulation (TOMACS) 14*(1), 76–114.

Alexopoulos, C. and A. F. Seila (1996). Implementing the batch means method in simulation experiments. In *Proceedings of the 28th Conference on Winter Simulation*, WSC '96, Washington, DC, USA, pp. 214–221. IEEE Computer Society.

An, G. and U. Wilensky (2009). From artificial life to in silico medicine. In M. Komosinski and A. Adamatzky (Eds.), *Artificial Life Models in Software*, pp. 183–214. London: Springer London.

Barde, S. (2016). Direct comparison of agent-based models of herding in financial markets. *Journal of Economic Dynamics and Control 73*, 329–353.

Belzner, L., R. De Nicola, A. Vandin, and M. Wirsing (2014). Reasoning (on) service component ensembles in rewriting logic. In *Specification, Algebra, and Software*, Volume 8373 of *LNCS*, pp. 188–211. Springer.

Belzner, L., R. Hennicker, and M. Wirsing (2016). Onplan: A framework for simulation-based online planning. In *Formal Aspects of Component Software*, Volume 9539 of *LNCS*, pp. 1–30. Springer.

Beygelzimer, A., J. Langford, and D. Pennock (2012). Learning performance of prediction markets with kelly bettors. *arXiv preprint arXiv:1201.6655*.

Billingsley, P. (1995). *Probability and measure*. John Wiley & Sons.

Bortolussi, L., D. Milios, and G. Sanguinetti (2015). Machine learning methods in statistical model checking and system design - tutorial. In E. Bartocci and R. Majumdar (Eds.), *Runtime Verification - 6th International Conference, RV 2015 Vienna, Austria, September 22-25, 2015. Proceedings*, Volume 9333 of *Lecture Notes in Computer Science*, pp. 323–341. Springer.

Bottazzi, G. and D. Giachini (2017). Wealth and price distribution by diffusive approximation in a repeated prediction market. *Physica A: Statistical Mechanics and its Applications 471*, 473–479.

Bottazzi, G. and D. Giachini (2019a). Betting, selection, and luck: A long-run analysis of repeated betting markets. *Entropy 21*(6), 585.

Bottazzi, G. and D. Giachini (2019b). Far from the madding crowd: Collective wisdom in prediction markets. *Quantitative Finance 19*(9), 1461–1471.

Brown, D. G., S. Page, R. Riolo, M. Zellner, and W. Rand (2005). Path dependence and the validation of agent-based spatial models of land use. *International Journal of Geographical Information Science 19*(2), 153–174.

Caiani, A., A. Godin, E. Caverzasi, M. Gallegati, S. Kinsella, and J. E. Stiglitz (2016). Agent based-stock flow consistent macroeconomics: Towards a benchmark model. *Journal of Economic Dynamics and Control 69*, 375–408.

Caiani, A., A. Russo, and M. Gallegati (2019). Does inequality hamper innovation and growth? an ab-sfc analysis. *Journal of Evolutionary Economics 29*(1), 177–228.

Carley, K. M., D. B. Fridsma, E. Casman, A. Yahja, N. Altman, L.-C. Chen, B. Kaminsky, and D. Nave (2006). Biowar: scalable agent-based model of bioattacks. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 36*(2), 252–265.

Chow, S.-C., J. Shao, and H. Wang (2002). A note on sample size calculation for mean comparisons based on noncentral t-statistics. *Journal of biopharmaceutical statistics 12*(4), 441–456.

Ciancia, V., D. Latella, M. Massink, R. Paškauskas, and A. Vandin (2016). A tool-chain for statistical spatio-temporal model checking of bike sharing systems. In *Leveraging Applications of Formal Methods*, Volume 9952 of *LNCS*, pp. 657–673.

Cincotti, S., M. Raberto, and A. Teglio (2010). Credit money and macroeconomic instability in the agent-based model and simulator eurace. *Economics: The Open-Access, Open-Assessment E-Journal 4*.

Cohen, J. (1992). A power primer. *Psychological bulletin 112*(1), 155.

Conway, R. W. (1963). Some tactical problems in digital simulation. *Management Science 10*(1), 47–61.

Dahlke, J., K. Bogner, M. Mueller, T. Berger, A. Pyka, and B. Ebersberger (2020). Is the juice worth the squeeze? machine learning (ml) in and for agent-based modelling (abm).

Dawid, H., S. Gemkow, P. Harting, S. Van der Hoog, and M. Neugart (2012). The eurace@ unibi model: An agent-based macroeconomic model for economic policy analysis.

Delli Gatti, D., C. Di Guilmi, E. Gaffeo, G. Giulioni, M. Gallegati, and A. Palestrini (2005). A new approach to business fluctuations: heterogeneous interacting agents, scaling laws and financial fragility. *Journal of Economic behavior & organization 56*(4), 489–512.

Delli Gatti, D. and J. Grazzini (2020). Rising to the challenge: Bayesian estimation and forecasting techniques for macroeconomic agent based models. *Journal of Economic Behavior & Organization 178*, 875–902.

Dosi, G., G. Fagiolo, M. Napoletano, A. Roventini, and T. Treibich (2015). Fiscal and monetary policies in complex evolving economies. *Journal of Economic Dynamics and Control 52*(C), 166–189.

Dosi, G., G. Fagiolo, and A. Roventini (2010). Schumpeter meeting keynes: A policy-friendly model of endogenous growth and business cycles. *Journal of Economic Dynamics and Control 34*(9), 1748–1767.

Dosi, G. and A. Roventini (2019). More is different... and complex! the case for agent-based macroeconomics. *Journal of Evolutionary Economics 29*(1), 1–37.

Dosi, G., A. Roventini, and E. Russo (2019). Endogenous growth and global divergence in a multi-country agent-based model. *Journal of Economic Dynamics and Control 101*, 101–129.

Effken, J. A., K. M. Carley, J.-S. Lee, B. B. Brewer, and J. A. Verran (2012). Simulating nursing unit performance with orgahead: strengths and challenges. *Computers, informatics, nursing: CIN 30*(11), 620.

Fagiolo, G., D. Giachini, and A. Roventini (2020). Innovation, finance, and economic growth: an agent-based approach. *Journal of Economic Interaction and Coordination 15*(3), 703–736.

Fagiolo, G., M. Guerini, F. Lamperti, A. Moneta, and A. Roventini (2019). Validation of agent-based models in economics and finance. In *Computer Simulation Validation*, pp. 763–787. Springer.

Fagiolo, G. and A. Roventini (2012). Macroeconomic policy in dsge and agent-based models. *Revue de l'OFCE 124*, 67–116.

Fagiolo, G. and A. Roventini (2017). Macroeconomic policy in dsge and agent-based models redux: New developments and challenges ahead. *Journal of Artificial Societies and Social Simulation 20*(1).

Franke, R. and F. Westerhoff (2012). Structural stochastic volatility in asset pricing dynamics: Estimation and model contest. *Journal of Economic Dynamics and Control 36*(8), 1193–1211.

Gibbons, J. D. (1986). Nonparametric statistical inference, 2nd. ed. statistics: Textbooks and monographs vol. 65. marcel dekker, inc., new york and basel 1985, xv, 408 s., $ 41,25 ($ 34,50 us and canada). *Biometrical Journal 28*(8), 936–936.

Gilmore, S., D. Reijsbergen, and A. Vandin (2017). Transient and steady-state statistical analysis for discrete event simulators. In *Integrated Formal Methods - 13th International Conference, IFM 2017, Turin, Italy, September 20-22, 2017, Proceedings*, pp. 145–160.

Gilmore, S., M. Tribastone, and A. Vandin (2014). An analysis pathway for the quantitative evaluation of public transport systems. In *Integrated Formal Methods*, Volume 8739 of *LNCS*, pp. 71–86. Springer.

Godley, W. and M. Lavoie (2006). *Monetary economics: an integrated approach to credit, money, income, production and wealth*. Springer.

Gray, R. M. (2009). *Probability, random processes, and ergodic properties*, Volume 1. Springer.

Grazzini, J. (2012). Analysis of the emergent properties: Stationarity and ergodicity. *Journal of Artificial Societies and Social Simulation 15*(2), 7.

Grazzini, J. and M. Richiardi (2015). Estimation of ergodic agent-based models by simulated minimum distance. *Journal of Economic Dynamics and Control 51*, 148–165.

Grazzini, J., M. G. Richiardi, and M. Tsionas (2017). Bayesian estimation of agent-based models. *Journal of Economic Dynamics and Control 77*, 26–47.

Grimm, V. and S. F. Railsback (2013). *Individual-based modeling and ecology*. Princeton university press.

Guerini, M. and A. Moneta (2017). A method for agent-based models validation. *Journal of Economic Dynamics and Control 82*, 125–141.

Ilachinski, A. (1997). Irreducible semi-autonomous adaptive combat (isaac): An artificial-life approach to land warfare. Technical report, DTIC Document.

Kelton, W. D. and A. M. Law (1984). An analytical evaluation of alternative strategies in steady-state simulation. *Oper. Res. 32*(1), 169–184.

Kets, W., D. M. Pennock, R. Sethi, and N. Shah (2014). Betting strategies, market selection, and the wisdom of crowds. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Kukacka, J. and L. Kristoufek (2020). Do 'complex'financial models really lead to complex dynamics? agent-based models and multifractality. *Journal of Economic Dynamics and Control 113*, 103855.

Lada, E. K., A. C. Mokashi, and J. R. Wilson (2013). Ard: An automated replication-deletion method for simulation analysis. In *2013 Winter Simulations Conference (WSC)*, pp. 802–813. IEEE.

Lamperti, F. (2018a). Empirical validation of simulated models through the gsl-div: an illustrative application. *Journal of Economic Interaction and Coordination 13*(1), 143–171.

Lamperti, F. (2018b). An information theoretic criterion for empirical validation of simulation models. *Econometrics and Statistics 5*, 83–106.

Lamperti, F., V. Bosetti, A. Roventini, and M. Tavoni (2019). The public costs of climate-induced financial instability. *Nature Climate Change 9*(11),

829–833.
Lamperti, F., G. Dosi, M. Napoletano, A. Roventini, and A. Sapio (2018). Faraway, so close: Coupled climate and economic dynamics in an agent-based integrated assessment model. *Ecological Economics 150*, 315 – 339.
Lamperti, F., G. Dosi, M. Napoletano, A. Roventini, and A. Sapio (2020). Climate change and green transitions in an agent-based integrated assessment model. *Technological Forecasting and Social Change 153*, 119806.
Lamperti, F., A. Roventini, and A. Sani (2018). Agent-based model calibration using machine learning surrogates. *Journal of Economic Dynamics and Control 90*, 366–389.
Law, A. M. and J. S. Carson (1979). A sequential procedure for determining the length of a steady-state simulation. *Operations Research 27*(5), 1011–1025.
Law, A. M. and D. M. Kelton (2015). *Simulation Modeling and Analysis* (5th ed.). McGraw-Hill Higher Education, http://www.averill-law.com/simulation-book/.
L'Ecuyer, P. (2016). SSJ: Stochastic simulation in Java, software library. http://simul.iro.umontreal.ca/ssj/.
L'Ecuyer, P., L. Meliani, and J. Vaucher (2002). SSJ: A framework for stochastic simulation in Java. In E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes (Eds.), *Proceedings of the 2002 Winter Simulation Conference*, pp. 234–242. IEEE Press.
Lee, J. S., T. Filatova, A. Ligmann-Zielinska, B. Hassani-Mahmooei, F. Stonedahl, I. Lorscheid, A. Voinov, G. Polhill, Z. Sun, and D. C. Parker (2015). The complexities of agent-based modeling output analysis. *The journal of artificial societies and social simulation 18*(4).
Legay, A., A. Lukina, L. Traonouez, J. Yang, S. A. Smolka, and R. Grosu (2019). Statistical Model Checking. In B. Steffen and G. J. Woeginger (Eds.), *Computing and Software Science: State of the Art and Perspectives*, Volume 10000 of *LNCS*, pp. 478–504. Springer.
Legay, A., S. Sedwards, and L. Traonouez (2016). Rare events for statistical model checking an overview. In K. G. Larsen, I. Potapov, and J. Srba (Eds.), *Reachability Problems - 10th International Workshop, RP 2016, Aalborg, Denmark, September 19-21, 2016, Proceedings*, Volume 9899 of *Lecture Notes in Computer Science*, pp. 23–35. Springer.
Lehr, R. (1992). Sixteen s-squared over d-squared: A relation for crude sample size estimates. *Statistics in medicine 11*(8), 1099–1102.
Lux, T. and R. C. Zwinkels (2018). Empirical validation of agent-based models. In *Handbook of computational economics*, Volume 4, pp. 437–488. Elsevier.
Macy, M. W. and R. Willer (2002). From factors to actors: Computational sociology and agent-based modeling. *Annual review of sociology*, 143–166.
Mandes, A. and P. Winker (2017). Complexity and model comparison in agent based modeling of financial markets. *Journal of Economic Interaction and Coordination 12*(3), 469–506.
Pianini, D., S. Sebastio, and A. Vandin (2014). Distributed statistical analysis of complex systems modeled through a chemical metaphor. In *International Conference on High Performance Computing & Simulation, HPCS 2014, Bologna, Italy, 21-25 July, 2014*, pp. 416–423.
Popoyan, L., M. Napoletano, and A. Roventini (2020). Winter is possibly not coming: Mitigating financial instability in an agent-based model with interbank market. *Journal of Economic Dynamics and Control*, 103937.
Richiardi, M. G., R. Leombruni, N. J. Saam, and M. Sonnessa (2006). A common protocol for agent-based social simulation. *Journal of artificial societies and social simulation 9*.
Richiardi, M. G. and R. E. Richardson (2017). Jas-mine: A new platform for microsimulation and agent-based modelling. *International Journal of Microsimulation 10*(1), 106–134.
Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. *Biometrics bulletin 2*(6), 110–114.
Sebastio, S. and A. Vandin (2013). Multivesta: statistical model checking for discrete event simulators. In *7th International Conference on Performance Evaluation Methodologies and Tools, ValueTools '13, Torino, Italy, December 10-12, 2013*, pp. 310–315.
Secchi, D. and R. Seri (2017). Controlling for false negatives in agent-based models: a review of power analysis in organizational research. *Computational and Mathematical Organization Theory 23*(1), 94–121.
Sen, K., M. Viswanathan, and G. Agha (2004). Statistical model checking of black-box probabilistic systems. In *CAV 2004*, pp. 202–215. Springer.
Seri, R. and D. Secchi (2017). How many times should one run a computational simulation? In *Simulating Social Complexity*, pp. 229–251. Springer.
Steiger, N. M., E. K. Lada, J. R. Wilson, J. A. Joines, C. Alexopoulos, and D. Goldsman (2005). Asap3: A batch means procedure for steady-state simulation analysis. *ACM Transactions on Modeling and Computer Simulation (TOMACS) 15*(1), 39–73.
Steiger, N. M. and J. R. Wilson (2001). Convergence properties of the batch means method for simulation output analysis. *INFORMS Journal on Computing 13*(4), 277–293.
Stuart, A. and J. K. Ord (1994). *Kendall's Advanced Theory of Statistics, Volume 1: Distribution Theory* (6th ed.).
Tafazzoli, A., J. R. Wilson, E. K. Lada, and N. M. Steiger (2011). Performance of skart: A skewness-and autoregression-adjusted batch means procedure for simulation analysis. *INFORMS Journal on Computing 23*(2), 297–314.
ter Beek, M. H., A. Legay, A. L. . Lafuente, and A. Vandin (2020). A framework for quantitative modeling and analysis of highly (re) configurable systems. *IEEE Trans. Software Eng. 46*(3), 321–345.
ter Beek, M. H., A. Legay, A. Lluch-Lafuente, and A. Vandin (2015). Quantitative analysis of probabilistic models of software product lines with statistical model checking. In *Proceedings 6th Workshop on Formal Methods and Analysis in SPL Engineering, FMSPLE@ETAPS 2015, London, UK, 11 April 2015*, pp. 56–70.
Tesfatsion, L. and K. L. Judd (2006). *Handbook of computational economics: agent-based computational economics*. Elsevier.
Valente, M. (2008). Laboratory for simulation development: Lsd. Technical report, LEM Working Paper Series.
van der Hoog, S. (2019). Surrogate modelling in (and of) agent-based models: A prospectus. *Computational Economics 53*(3), 1245–1263.
von Neumann, J. (1941). Distribution of the ratio of the mean square successive difference to the variance. *Ann. Math. Statist. 12*(4), 367–395.
Wald, A. and J. Wolfowitz (1940). On a test whether two samples are from the same population. *The Annals of Mathematical Statistics 11*(2), 147–162.
Welch, B. L. (1947). The generalization ofstudent's' problem when several different population variances are involved. *Biometrika 34*(1/2), 28–35.
Welch, P. D. (1983). The statistical analysis of simulation results. *The computer performance modeling handbook 22*, 268–328.
Whitt, W. (1991). The efficiency of one long run versus independent replications in steady-state simulation. *Management Science 37*(6), 645–666.

Winker, P., M. Gilli, and V. Jeleskovic (2007). An objective function for simulation based inference on exchange rate data. *Journal of Economic Interaction and Coordination 2*(2), 125–145.
Younes, H. L. (2005). Probabilistic verification for "black-box" systems. In *CAV 2015*, pp. 253–265. Springer.