A Toolbox to Analyze Emergence in Multiagent Simulations

Danilo Saft and Volker Nissen

Additional information is available at the end of the chapter

Abstract

The field of complexity science often employs multiagent simulations to investigate complex and emergent behavior. Authors in complexity science have suggested that the discussion of complex systems could benefit from a more systematic approach and a more compact mathematical way to describe the behavior of such systems in addition to the common observations and interpretations taking place today. Regarding quantitative measures to capture emergent phenomena, several approaches have been published, but have not yet been put to wide systematic use in the research community. One reason for this could be the manual effort required to investigate multiagent systems in a quantitatively accurate form. Toward this end, there has so far been a lack of appropriate and easy-to-use IT-based tools. To eliminate this deficiency, we present a software library, which enables researchers to integrate emergence measurements into experiments with multiagent modeling tools such as Repast and NetLogo. The major benefit for researchers is that this toolbox enables them to make comparable, quantitatively well-grounded statements about the emergent behavior of the model at hand. The toolbox therefore provides researchers with a standardized artifact that can be employed in a systematic methodological approach to the analysis of multiagent systems.

Keywords: multiagent system, emergence measures, complexity mathematics, toolbox, agent-based simulation, evaluation

1. Brief introduction, problem statement, and research goals

The field of complexity science is interdisciplinary and extends broadly into areas of science such as physics, biology, sociology, economics, and others [4]. Agent-based systems can be considered as computational laboratories [5–7] that provide research insights analogously to conventional laboratories. The research community usually concentrates investigations on the study of the system behavior and on the relation of input and output variables within multiagent simulations. Authors in complexity science have suggested that the discussion of

complex systems could benefit from a more systematic approach and a more compact mathematical way to describe the behavior of such systems in addition to the common observations and interpretations taking place today ([1], pp. 233–235). Ways to measure macroscopic features of complex agent-based systems, such as emergence, were already presented in the literature [2, 3]. However, complexity phenomena and emergence, when subjectively observed, thus far have mostly been described qualitatively in publications on multiagent systems. Yet, emergence measures provide an elegant and significant opportunity to verify accurately the notion of emergence taking place in a multiagent simulation, for example, in self-organization processes. With the measure of Granger-Emergence [3], researchers can even quantify on a metric scale, the intensity of emergence within a process. The manual effort, that is, the number of manual calculations or human–computer interactions in the form of separate commands, necessary to measure emergence may make it difficult to include this dimension in one's research without being supported by an easy-to-use computer-aided process. Software libraries to measure Granger-Emergence via the measure of Granger-causality exist [8], however, up until now only as a program library for the software suite Matlab and with an extensive interface. Thus, our objective is to provide the scientific community with a publicly available IT-based tool to measure emergence in agent-based simulations. Besides broader access to such software libraries, the automation of emergence measurements and its quick integration with multiagent simulation frameworks are our further goals in this endeavor. A large part of the scientific community works with free simulation and analysis programs such as NetLogo [9], Repast Simphony [10], and R [11]. Guides on integrating these tools have been published, for example, in Ref. [12] and [13]. Consequently, the easy-to-use toolbox to accurately measure emergence, we aim to provide uses of the R framework. The integration of our library in experiments with broadly used multiagent simulation frameworks, such as Netlogo, will be demonstrated in this chapter. Here, we will use a standard model of bird flocking behavior to demonstrate the use of the toolbox toward the quantitative analysis of emergent systems.

2. Research method and structure

A toolbox represents an artifact as defined by Hevner et al. [14]. Therefore, the design science research (DSR) approach information systems science provides an adequate methodology for our project. Using design science research, one is able to systematically create IT artifacts in order to find solutions to practical problems. Particularly, we are guided by the well-known DSR process model by Peffers et al. ([15], p. 54), which organizes the design process iteratively and structures it in phases as shown in **Figure 1** and briefly described below.

The artifact we wish to create is an IT-based toolbox that enables researchers to easily measure emergence within their multiagent simulations. It therefore provides an opportunity to extract more quantitative information about properties such as emergence within multiagent systems. We will explain the designed artifact and demonstrate its application to an exemplary multiagent system. We conclude with an initial discussion of the results of applying the artifact as a starting point for its evaluation. This publication represents our attempt to communicate our findings to the scientific community.
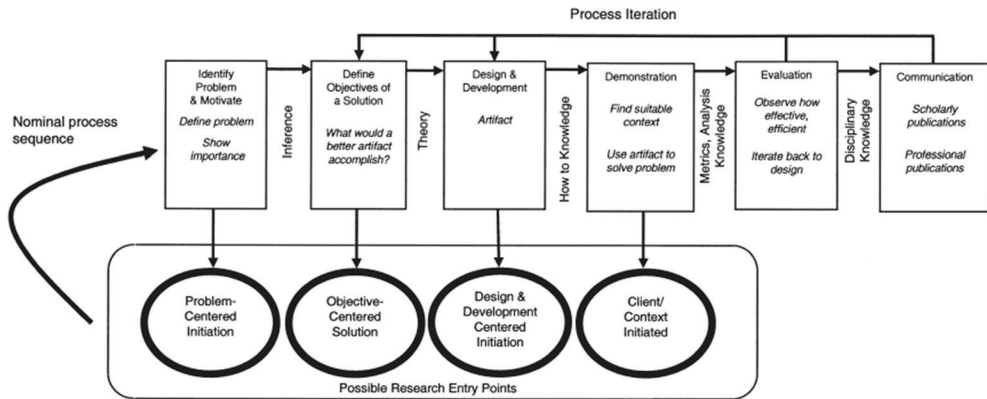
Figure 1. Design science research process model by Peffers et al. ([15], p. 54).

Hence, the structure of our book chapter is as follows: We first present the problem and the objective for a possible solution as shown in the introductory segment. By explaining the fundamentals of Granger-Emergence as a measure for weak emergence, we provide the foundation to understand the usability and benefits of the tool we design and develop. Then, we will design and present the artifact in the form of an R toolbox and its integrated utilization from typical NetLogo simulation results. The applicability and use of this tool is demonstrated with an exemplary simulation of flocks of birds (boids), measuring the emergence of flocks on the macrolevel of the simulation over the movements of each boid on the microlevel of the simulation. The measurements will be taken under different environmental parameter settings and emergence will be analyzed for each setting. Finally, we discuss positive and negative insights gained by applying the toolbox to the simulation model and conclude with some ideas for future research and tool extensions.

3. Foundations: measuring Granger-Emergence

Emergence is a phenomenon that occurs in complex systems when macroscopic processes of the system arise from the concurrence of microscopic processes, while at the same time, the macroscopic process cannot be reduced to just the sum of its constituent microprocesses. Each particular macroprocess can be seen to emerge only by simulation/full calculation and not be simply deduced prior to the simulation of all microinteractions. Notably, the occurrence of synchronous weak emergence [16] is of particular interest to the research community. Seth transformed [3] Bedaus approach to weak emergence [17] into a measure for weak synchronous emergence, the so-called Granger-Emergence. Hence, emergence can be measured via a time series analysis of a macrovariable in (Granger-causal) relation to its microvariables. This picks up Bedaus notion [17] that a weakly emergent process is both dependent upon its microcausal influences and at the same time autonomous from them. Seth's approach to this notion is to find out whether "(i) past observations of [the macrovariable] help predict future

observations of [the macrovariable] with greater accuracy than predictions based on past observations of [the underlying set of microvariables] alone, and (ii) past observations of [the microvariables] help predict future observations of [the macrovariable] with greater accuracy than predictions based on past observations of [the macrovariable] alone.". This means that in order to find out whether synchronous weak emergence is present in a particular time step of a multiagent simulation, researchers need to answer the following questions:

a. Does the knowledge about a (more precise: each) single *agent's* parameter in *past* time-steps in the simulation give an advantage in (statistically) deriving the *current* measurement of the observed macroparameter thought to represent an *emergent phenomenon*?

b. Does the knowledge about *past* observations of the macroparameter representing the *emergent phenomenon* also help to predict the *current* value of the said parameter?

To answer these questions, researchers first need to capture the microlevel observations (i.e., parameters of each agent) as well as a quantified measure for the macrolevel phenomenon in a matrix of time series for each observed parameter. When simulating a boids model, this would imply recording the position of each simulated bird in each time step as well as recording a measure of the degree of flocking. Simple measures could be the standard deviation of positions, or the mean of squared numbers of other boids in the field of sight of the bird. More sophisticated and performance-demanding measures would be numbers of clusters found by a cluster analysis on the positions of each boid in the current time step of the simulation or other metrics of clustering and segregation derived from distance measures. Examples of this step will be explained in Section 5.

After recording the necessary data, question (*a*) from above can be quantitatively answered by calculating the so-called *Granger-causality* for the recorded macrolevel observations over the microlevel observations. To this end, one builds a (multivariate) statistical model for the explanation of the current macrovariable by the past observations. Seth [3] explains this by employing a demonstration example of a two-variable autoregressive process (Eqs. (1a) and (1b)). Here, we assume $X_1(t)$ is the time series for the macrovariable and $X_2(t)$ is the time series for a single microvariable parameter of the simulation; with t denoting the time-step of the simulation for which the values were recorded.

$$X_1(t) = \sum_{j=1}^{p} A_{11,j} X_1(t-j) + \sum_{j=1}^{p} A_{12,j} X_2(t-j) + \varepsilon_1(t) \tag{1a}$$

$$X_2(t) = \sum_{j=1}^{p} A_{21,j} X_1(t-j) + \sum_{j=1}^{p} A_{22,j} X_2(t-j) + \varepsilon_2(t) \tag{1b}$$

Here, p is the maximum number of lags of the variables, that is, the number of time-steps in the past of the simulation, for which observations of each variable are taken into consideration for the calculation of the current value of the parameter, thus denoting the duration for which the past observations are thought to influence the present observations. A_{ab} represents a matrix for each term that contains the coefficients for each current output variable $X_a(t)$ explained by lagged observations $X_b(t-j)$. ε_a represents the residual error term of each time series process and will be of particular interest in answering the questions posed above.

To clarify further, using a short example, assume we observe only two variables and know the last two values of these variables (i.e., lag 2). The autoregressive process notations resulting from this example would equate to

$$X_1(t) = c_{111}X_1(t-1) + c_{112}X_1(t-2) + c_{121}X_2(t-1) + c_{122}X_2(t-2) + \varepsilon_1(t), \qquad (2)$$

representing a hypothetical "macrolevel observation" X_1 and

$$X_2(t) = c_{211}X_1(t-1) + c_{212}X_1(t-2) + c_{221}X_2(t-1) + c_{222}X_2(t-2) + \varepsilon_2(t), \qquad (3)$$

representing a hypothetical single microlevel observation X_2.[1]

In this case, we assume we want to test whether $X_1(t)$, that is, the observed values of the macrovariable, are Granger-caused by the past (here: two) observations of the micro-variables $X_2(t - j)$. X_2 Granger-causes X_1 if the inclusion of the $X_2(t - j)$ terms in Eq. (2) leads to *better prediction* of X_1 than the exclusion does. Better prediction in turn is indicated by a lower variance of the residual term $\varepsilon_1(t)$. Hence, we first construct an autoregressive model that is *restricted* in comparison to the unrestricted model in Eq. (2):

$$X_{1R2}(t) = c_{111}X_1(t-1) + c_{112}X_1(t-2) + \varepsilon_{1R2}(t). \qquad (4)$$

The model in Eq. (4) tries to predict $X_1(t)$ while leaving out information on X_2. Granger-causality is then the comparison of the variance of the residual values of the restricted model versus the unrestricted model [3]:

$$gc_{X2\to X1} = log\frac{var(\varepsilon_{1R2})}{var(\varepsilon_1)} . \qquad (5)$$

More details regarding the calculation of Granger-causality can be found in Ref. ([3], pp. 545–547).

Seth introduces a derivate of Granger-causality that he denotes as Granger-autonomy ([3], p. 547). This derivate is able to answer the yet unaddressed question *b* posed above in this section. Question b regards the influence of the macrovariables own past on the predictability of its current state. Modifying the example given in Eq. (4), Granger-autonomy is a calculation of Granger-causality applied to lagged observations of X_1. Accordingly, the restricted model equates to

$$X_{1R1}(t) = c_{121}X_2(t-1) + c_{122}X_2(t-2) + \varepsilon_{1R1}(t). \qquad (6)$$

Granger-autonomy of process X_1, as posed by ([3], p. 547), therefore equates to

---

[1]Here, we initially ignore the circumstance that practical attempts to measure emergence via Granger-causality will later include more than one micro-variable. This example is just to demonstrate the step-by-step construction of the equations and calculations underlying the measurement process.

$$ga_{X1} = g_{c|X1, X1} = \log \frac{var(\varepsilon_{1R1})}{var(\varepsilon_1)}. \tag{7}$$

Weak synchronous emergence can then be calculated by what Seth [3] called Granger-Emergence, by putting the Granger-autonomy value of a macroprocess (here: X_1) in relation to the average of all Granger-causalities of the microprocesses (here: only X_2) toward the macroprocess:

$$ge_{X1|X2} = ga_{X1} * \frac{1}{1} \left(\Sigma_{i=2}^{i=2} gc_{X_i \to X1} \right) = \log \frac{var(\varepsilon_{1R1})}{var(\varepsilon_1)} * \log \frac{var(\varepsilon_{1R2})}{var(\varepsilon_1)}. \tag{8}$$

4. Design and development of the toolbox

From the short example in Eqs. (2) and (3), it becomes apparent that while the operations are simple, the number of calculations to be performed can rise to an amount unsuitable for even just semimanual calculation of autoregressive models in multiagent simulations, where the number of observed microvariables can be large compared to just a single microvariable $X_2(t)$. Additionally, in larger agent-based models with a variety of variables to be evaluated for their link to a macrophenomenon, the determination of Granger-Emergence can be time consuming both with regard to data-preparation and separate calculation steps that are to be carried out by the researcher. Consequently, it is our goal to create an IT artifact that is capable of largely automating all calculation steps to compute the following:

- Derive a vector of Granger-causalities for each microparameter toward a given macro-variable.

- Determine the Granger-autonomy of the macrolevel observations given the simulation data.

- Calculate a Granger-Emergence value given just the simulation data and the macrovariable name.

Additionally, the following functional objectives should be accomplished:

- Easy setup and integration into existing and new models of multiagent systems.

- Flexible calculations for both the end-result sets of experimental data as well as live calculations as the simulation progresses.

- Provide researchers with quantitative measures of emergence, making it possible to compute the following:

  o Compare the amount of emergence in parameter studies of the model, finding peaks of emergent phenomena. This makes it possible to find parameter ranges for the model to promote or avoid emergence.

  o Give quantitative evidence for formerly qualitative interpretations of emergence phenomena based on a standard measure and a standard tool.

- Avoid calculation errors (quality assurance).

- It must serve as a basis for further extensions that help researchers in the field of complexity science: for example, the toolbox may be able to later automate the process of emergence

detection to a level where it is possible to just call a function with a given model and have the toolbox permute through all variables automatically to find possible emergence patterns.

• Save a significant amount of time on quantifying synchronous weak emergence for one's models.

In this section, we will first present a self-developed set of functions for the well-known statistical programming language *R*. This toolbox provides methods to analyze R dataframes for Granger-Emergence. Dataframes are a standardized structure in R that will, in our case, hold experimental results of an agent-based simulation in the form of time series values for each model variable. Given a dataframe with the simulation results and the name of the column that contain the macrolevel observations, the toolbox will be automatically capable of calculating Granger-Emergence for the simulation data. Researchers may also customize the function call to look only for emergence in specific combinations of the model variables or specific lags in the history of the simulation data that are to be considered.

Regarding the design of a toolbox, the first key feature that enables calculation of Granger-Emergence needs is an ability to construct a formula for an unrestricted model of a time-series, in which the macrovariable is explained by lagged observation of all other variables. It also needs to create a restricted model formula, which excludes a given microvariable from the unrestricted version of the formula. The R-code to derive such a model formula is given below:

```
#create a formula object using the string output of the functions inside the bracket:

formula_restricted <- as.formula(

    #concatenate the comma-separated terms inside the paste0 function:

    paste0(

      macro_variable,    #variable name of the macro-observations

    "~",                 #="is to be explained by"

    paste0(              #concatenate the microvariables

      "lag(",            #Include Lags for each microvariable

        names(dataframe)[!names(dataframe) %in% restriction_variables],
#use each column except for the ones given in "restriction_variables"

        ", 1:", #create a lag sequence from 1 (first lag)

        lag_count, #to the defined lag count

      ")",

      collapse = "+" #add each lagged term to the model formula

    )

  )

 )
```

The benefit of this particular code is the utilization of Rs built-in vector-calculation routines. Apart from giving a better performance by avoiding nested for-loops, this makes the creation of the formula very flexible and elegant in that a dataframe of any size and any lag count may be given to derive a suitable formula of arbitrary length with very short, yet comprehensible, code. Moreover, the code offers adaptability for the microvariables to be tested for Granger-causality, since a vector of names can be assigned to restriction_variables. As an example for the creation of the regression formula, we can consider a dataframe with one macrovariable X1 and four microvariables X2..X5. We would like to consider the last 2 observations and calculate the Granger-causality of X4 toward X1. Given macro_variable <- "X1", lag_count <- 2 and restriction_variables <- "X4", the above code generates the following base formula for the restricted regression model:

$$X_1(t) \sim aX_1(t-1) + bX_1(t-2) + cX_2(t-1) + dX_2(t-2) + eX_3(t-1)$$

$$+ fX_3(t-2) + gX_5(t-1) + hX_5(t-2). \tag{9}$$

Note that the coefficients a to h will be determined by the regression method. Because the number of regression terms directly depends on the number of microvariables, it will be important to choose the number of simulations steps in the simulation greater than the number of terms in the constructed formula, including the lagged terms. This is because otherwise the model will be able to "overfit" the time-series and create residuals of zero. Zero-valued residuals make it impossible to calculate Granger-causality and can be a hint to such an implausibility in the simulation setup. To create a simple linear regression model, one could use the R function "lm(…)". Here, we utilize the function dynlm of the dynlm package [18] for didactic reasons, because it is particularly easy to update the restricted model to an unrestricted model using this package.

```
require(dynlm)
```

```
lm_restricted <- dynlm(formula = formula_restricted, data=dataframe)
```

```
lm_unrestricted <- update.dynlm(.~.+ lag(restriction_variables, lag_count), data=dataframe)
```

The code above generates a linear model based on the restricted model formula. The update command adds the omitted variable terms to retrieve an unrestricted model.

Both models are large objects with several nested properties. One of which is the $residuals object, giving a time series with the error terms of the fitted model in comparison to the actual values of the observations (here: of X_1) in the dataframe. Using these properties, we can calculate Granger-causality in R via

```
gc <- log(var(lm_restricted$residuals)/var(lm_unrestricted$residuals))
```

Together with a number of checks and convenience conversions for readability and adaptability, the toolbox is now able to provide an easy-to-use interface for calculating Granger-causalities for given macrovariables and microvariables in a dataframe:

```
grangerCausality <- function(dataframe, macro_variable, restriction_variables, lag_count) { … }
```

Using this interface, it is in turn easy to utilize it to calculate Granger-autonomy and Granger-Emergence:

```
grangerAutonomy <- function(dataframe, macro_variable, lag_count) {

    return(grangerCausality(dataframe, macro_variable, macro_variable, lag_count))

}

grangerEmergence <- function(dataframe, macro_variable, lag_count) {

    granger_causality_vector <- c()
    for (micro_variable in names(dataframe)) {

        if (micro_variable == macro_variable) {

         next

        }

        granger_causality_vector <- c(granger_causality_vector,
grangerCausality(dataframe, macro_variable, micro_variable, lag_count, lag_count))

    }

ga <- grangerAutonomy(dataframe, macro_variable, lag_count)

ge <- ga * 1/(ncol(dataframe) -1) * sum(granger_causality_vector)

return(ge)

}
```

Here, we calculate Granger-autonomy by applying the Granger-causality function to the macrovariable, omitting the lagged terms of the macroobservations in the restricted models. In the function Granger-Emergence, we first construct a vector of Granger-causalities for each microvariable in the dataframe to the macrovariable. Afterward, we calculate the Granger-autonomy of the macrovariable column in dataframe and return Granger-Emergence according to the formula provided in Ref. [3], applied to the particular dataframe provided as a parameter of the function call.

The complete code can be found on GitHub [19]. The codes will be updated further in upcoming versions with extensions listed in Section 6. A CRAN-package will be made available once the set of functions seems large enough.

5. Demonstrating an example

The R functions for calculating Granger-Emergence listed above provide sufficient functionality to demonstrate its general utilization and to validate the toolbox against a standard example. It is good practice to use widely known and widely available examples where possible.

We therefore build our demonstration upon the bird flocking model from the NetLogo example library.

In particular, we chose this model, due to the following reasons:

- Flocking is an apparent example of synchronous weak emergence.

- The NetLogo flocking model provides initialization parameters that make it possible to create seemingly random boid behavior or strong flocking behavior. This makes it possible to test the R code against contrasting results.

- A boid simulation was employed by Seth [3] to showcase the concept of Granger-Emergence. The general results of the demonstration in Ref. [3] should be roughly comparable to our findings.

A range of parameter instantiations of the bird flocking model were simulated. We used a population size of 15. The values for the parameter "vision" of each boid were set to 0 (no discovery of neighboring boids is possible) and to 9 (potentially leading to a strong influence by neighboring boids). Settings for the parameter "minimum separation" were cycled through the values 0.75, 1.5, 3.0, and 5.0. We simulated 1500 time steps in each run. Each simulation was performed 30 times. The settings can be automatically cycled and each combination saved to a CSV table using NetLogos integrated BehaviorSpace functions.

We modified the simulation code in NetLogo to be able to save the distance of each boid from the center and the standard deviation of the distance of all boids:

#BehaviorSpace value to save each boids distance and standard deviation

[distance patch 0 0] of turtles

rep_stddev

#report standard deviation of distances of boids

to-report rep_stddev

set xcorlist [distance patch 0 0] of turtles

set ycorlist [distance patch 0 0] of turtles

set stddev (0.5 * standard-deviation xcorlist) + (0.5 * standard-deviation ycorlist)

report stddev

end

The resulting CSV table of the BehaviorSpace experiments was read into an R dataframe omitting the first six rows and extracting only the necessary columns using the readr package [20]:

dataframe <- read_csv("./Flocking experiment-table_v0_ms0.75.csv", skip = 6)

The dataframe was transformed further to include 15 columns for the boid distance values in each step and 1 column for the macrovariable (here: standard deviation). A diff-log operation

was performed on each column to reach stationary behavior of the time-series in each column. These steps are in accordance with the preparation of data taken in the original experiment in Ref. [3]. Note that using the log function and taking first-order differences does not destroy the ability to investigate the data for Granger-Emergence correctly, since all necessary information is kept by the transformation. For performance reasons, we only considered lag 1 values in our regression model construction for the recorded time series. The Granger-Emergence measurement for each run of the simulation can be retrieved by the simple interface call below:

grangerEmergence(dataframe_filtered_by_run, "X16", 1)

For each of the 30 runs in a parameter combination, the overall Granger-Emergence value was saved into a separate dataframe containing the columns Vision, Minimum Separation and Granger-Emergence. Calculation for all given parameter combinations took about 30 min on an Intel Core i7 based standard pc. **Figure 2** shows the results in a boxplot comparison.

The chart shows the factor Vision on the x-axis. For each of the two Vision instances, four boxplots were plotted, indicating the settings for the parameter Minimum Separation. The y-axis shows the Granger-Emergence results of all 30 steps for each of the combinations as a boxplot, with outliers omitted. For a Vision value of zero, the median of the Granger-Emergence is close to zero for all settings of Minimum Separation. Of the number of outliers that were produced, only one of the outliers was particularly high-valued for vision 0 and was close to 0.9. One reason for this may be pseudorandom encounters of boids over several time steps that resembled flocking. However, and importantly, these false positives are single cases clearly and easily marked as outliers even in a small batch run series. The quartiles 1–3, that is, 75% of measurements, are very close to zero for a Vision value of zero. This is in stark contrast to emergence readings for a value of nine for the parameter Vision. For this setting, compelling flocking behavior was observed in the NetLogo simulation, when combined with low settings of Minimum Separation. Consequently, the median of the Granger-Emergence over all 30 runs is significantly higher as shown in **Figure 2**. Almost no outliers were produced for Vision = 9. The box, that is, half of the measurements, stretch over a spectrum of Granger-Emergence that is both wider and higher than in simulations where no flocking was observed. In accordance with the subjective observation of the simulations in
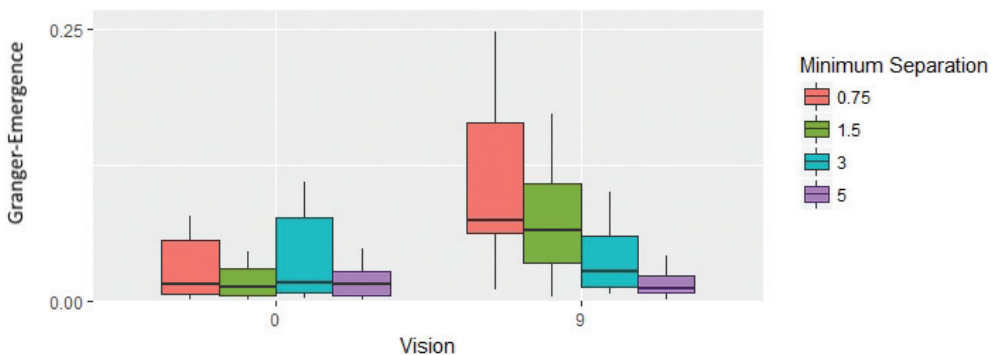

Figure 2. Boxplots of Granger-Emergence measurements over 30 runs each for parameter combinations of the NetLogo Flocking model.

NetLogo, the differences between Vision = 9 and 0 diminish when the Minimum Separation approaches approximately half of the Vision value.

6. Evaluation

The experimental results shown in this chapter are very comparable to those presented in Ref. [3]. While our specific experimental setup differs from Seth's boids simulations, we, too, obtain Granger-Emergence results that indicate the subjective observation of flocking behavior strikingly well. The methods presented above are suitable for distinguishing almost random behavior with little emergence from clearly emergent flocking behavior. Moreover, gradual differences in flocking behavior seem to leave a significant mark in the median of the measured emergence. We were able to validate our functions against the standard example of a NetLogo boids model and demonstrate its structure and usability. Very good and extensive implementations for the automated measurement of Granger-causality and Granger-autonomy exist for Matlab [8]. These implementations concentrate on a spectral perspective on Granger-causality, typically used in fields like neuroscience, and provide many interfaces for detailed tasks where this perspective fits well. Our approach successfully focused on providing a simple interface for researchers to measure directly the Granger-Emergence with a simple autoregressive modeling process according to Ref. [3]. Given the data from a simulation model, we provide a single function interface to receive immediate results in the open-source language R. This language offers direct integration with simulation tools such as Repast or NetLogo via interfaces like JRI [12]. A demonstration using this interface will be a topic for future articles of ours, building upon this publication.

The toolbox leaves room for extensions, improving convenience for researchers even further. For example, tests for significance of Granger-causality and Granger-autonomy may also be automated, saving time otherwise spent on semimanual calculation to verify the significance of the obtained results. Currently, the toolbox uses a linear regression modeling method for originally nonlinear time-series. While linear Granger-Emergence has been shown to resemble the nonlinear variants [3], we wish to provide a simple function interface to calculate nonlinear Granger-Emergence too. Furthermore, the package "dynlm" we currently employ utilizes only one processor core and may be slower to find regression models than other packages such as "speedglm" [21]. While the performance of the interface has already been optimized and is acceptable for typical models, further speed improvement is likely using GPU computation or multicore processing. All extensions will be maintained on the toolbox' GitHub project page [19] and will be accompanied by a series of articles demonstrating the particular solution. The toolbox is freely available to researchers already. With the extensions listed above, it will be able to provide extensive quantitative statements on multiagent systems and their parameter studies. We encourage further evaluation of the toolbox and discussion of improvements. Our hope is that its use will open a discussion on how to use quantification of subjectively emergent behavior in the discussion of multiagent simulations. To this end, we plan to publish a methodology for the systematic analysis of complex systems, incorporating the findings published in this chapter.

Author details

Danilo Saft* and Volker Nissen

*Address all correspondence to: danilo.saft@tu-ilmenau.de

Faculty of Economic Sciences and Media, Institute for Commercial Information Technology, Ilmenau Technical University, Ilmenau, Germany

References

[1] Miller JH, Page SE. Complex Adaptive Systems: An Introduction to Computational Models of Social Life. Princeton University Press; New Jersey; 2009

[2] Wright WA, Smith RE, Danek M, Greenway P. A generalisable measure of self-organisation and emergence. In: International Conference on Artificial Neural Networks; Springer; 2001. pp. 857–864

[3] Seth AK. Measuring emergence via nonlinear Granger causality. In: Bullock S, Watson R, Noble J, Bedau M, editors. Artificial Life XI: proceedings of the 11th international conference on the simulation and synthesis of living systems. MIT Press; Cambridge, Massachusetts; 2008. pp. 545–553

[4] Mitchell M. Complexity: A Guided Tour. Oxford University Press; New York; 2009

[5] Epstein JM, Axtell R. Growing Artificial Societies: Social Science from the Bottom Up. Brookings Institution Press; Washington; 1996

[6] Dibble C. Computational laboratories for spatial agent-based models. In: Handbook of Computational Economics. Vol. 2. Elsevier/North-Holland; Amsterdam; 2006. pp. 1511–1548

[7] Tesfatsion L. Agent-based computational economics: Modeling economies as complex adaptive systems. Information Sciences. 2003;**149**(4):262–268

[8] Barnett L, Seth AK. The MVGC multivariate Granger causality toolbox: A new approach to Granger-causal inference. Journal of Neuroscience Methods. 2014;**223**:50–68

[9] Wilensky U, Rand B. Introduction to Agent-Based Modeling: Modeling Natural, Social and Engineered Complex Systems with NetLogo. MIT Press; Cambridge, Massachusetts; 2015

[10] North MJ, Howe TR, Collier NT, Vos J. The repast Simphony runtime system. In: Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms; Argonne, IL. 2005

[11] R Core Team. R: A Language and Environment for Statistical Computing. 2013

[12] JRI. JRI - Java/R Interface [Internet]. JRI; (Website); 2013. Available from: http://www.webcitation.org/6NvIpV2bn [Accessed: 2013]

[13] Nissen V, Saft D. A practical guide for the creation of random number sequences from aggregated correlation data for multi-agent simulations. Journal of Artificial Societies and Social Simulation. 2014;**17**(4):7

[14] Hevner AR, March ST, Park J, Ram S. Design science in information system research. Management Information Systems Quarterly. 2004;**28**(1):75–105

[15] Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S. A design science research methodology for information systems research. Journal of Management Information Systems. 2008;**24**(3):45–77

[16] Stephan A, editors. "Emergenz." Von der Unvorhersagbarkeit zur Selbstorganisation. mentis; Dresden; 1999

[17] Bedau M. Weak emergence. Philosophical Perspectives. 1997: **11**, 375–399

[18] Zeileis A. Dynamic Linear Regression. 2016. Available from: http://CRAN.R-project.org/package=dynlm

[19] Saft D. Measure Emergence in R repository at GitHub. 2017. Available from: https://github.com/danilosaft/Measure-Emergence-in-R

[20] Hester J. Read Rectangular Text Data. 2017. Available from: https://cran.r-project.org/web/packages/readr/

[21] Enea M. Fitting Linear and Generalized Linear Models to Large Data Sets. 2017. Available from: https://cran.r-project.org/web/packages/speedglm/index.html