EvoDyn-3s: A Mathematica computable document to analyze evolutionary dynamics in 3-strategy games

Luis R. Izquierdo a, *, Segismundo S. Izquierdo b, William H. Sandholm c

a Universidad de Burgos, Department of Civil Engineering, Ed. A. Avda. Cantabria s/n, Burgos, 09006, Spain
b Universidad de Valladolid, Department of Industrial Organization, Paseo del Cauce 59, Valladolid, 47011, Spain
c University of Wisconsin, Department of Economics, University of Wisconsin, 1180 Observatory Drive, Madison, WI 53706, USA

ARTICLE INFO

Article history:
Received 23 May 2018
Received in revised form 18 July 2018
Accepted 18 July 2018

Keywords:
Evolutionary dynamics
Game theory
Mathematica
Phase portrait
Stability

ABSTRACT

EvoDyn-3s generates phase portraits of evolutionary dynamics, as well as data for the analysis of their equilibria. The considered evolutionary dynamics are ordinary differential equations based on adaptive processes taking place in a population of players who are randomly and repeatedly matched in couples to play a 2-player symmetric normal-form game with three strategies. EvoDyn-3s calculates the rest points of the dynamics using exact arithmetic, and represents them. It also provides the eigenvalues of the Jacobian of the dynamics at the isolated rest points, which are useful to evaluate their local stability. The user only needs to specify the 3 × 3 payoff matrix of the game and choose the dynamics.

© 2018 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Code metadata

Current code version v1.0
Permalink to code/repository used for this code version https://github.com/ElsevierSoftwareX/SOFTX_2018_63
Legal Code License GNU General Public License (GPL)
Code versioning system used GitHub
Software code languages, tools, and services used Mathematica
Compilation requirements, operating environments & dependencies Most EvoDyn-3s’s functionality can be used with the free Wolfram CDF Player. Full functionality requires Mathematica. Both Wolfram CDF Player and Mathematica run on Windows, Mac and Linux.
Documentation included within the computable document lrizquierdo@ubu.es
If available Link to developer documentation/ manual
Support email for questions lrizquierdo@ubu.es

Software metadata

Current software version v1.0
Permalink to executables of this version https://github.com/luis-r-izquierdo/EvoDyn-3s/releases/tag/v1.0
Legal Software License GNU General Public License (GPL)
Computing platforms/Operating Systems Any that can run Wolfram CDF files. E.g. Windows, Mac, Linux, and iOS.
Installation requirements & dependencies Most EvoDyn-3s’s functionality can be used with the free Wolfram CDF Player. Full functionality requires Mathematica.
Documentation included within the computable document lrizquierdo@ubu.es
If available, link to user manual - if formally published include a reference to the publication in the reference list
Support email for questions lrizquierdo@ubu.es

* Corresponding author.
E-mail addresses: lrizquierdo@ubu.es (L.R. Izquierdo), segis@eii.uva.es (S.S. Izquierdo), whs@ssc.wisc.edu (W.H. Sandholm).

https://doi.org/10.1016/j.softx.2018.07.006
2352-7110/© 2018 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
1. Motivation and significance

Social and biological interactions among agents who may adopt different actions are usually modeled as games. Frequently, these games are studied from an adaptive or evolutionary perspective, leading to the study of ordinary differential equations known as evolutionary game dynamics. A paradigmatic example is the replicator dynamics [1–4], which have become a standard reference case for analyzing adaptive processes in evolutionary game theory, color the background according to their speed (i.e. the modulus of the derivative vector), calculates the rest points, and provides the eigenvalues of the Jacobian at isolated rest points in order to analyze their local stability (in dynamics where the Jacobian is defined).

The user can analyze any $3 \times 3$ game by setting the values $a_{ij}$ of the $3 \times 3$ payoff matrix, which represent the payoff that a player using strategy $i \in \{1, 2, 3\}$ obtains when interacting with a player using strategy $j \in \{1, 2, 3\}$. There is also a list of predefined games that the user can choose from.

---

Fig. 1 shows the interface of EvoDyn-3s. The left part contains a series of input boxes and various controls that are used to set the values of all parameters. The right part shows the main output of the program: a phase portrait in the 2-dimensional simplex, a table showing all the isolated rest points and the eigenvalues of the Jacobian of the chosen dynamic at each of the isolated rest points (if the Jacobian is defined), and another table showing the components of rest points (if any exist).

The effect of changing the value of any parameter (except the payoff matrix) on the phase portrait and on the computation of rest points and eigenvalues is immediate, i.e. there is no need to compile or rerun the program. Thus, for example, the user can gradually move any parameter slider and immediately appreciate how this change affects the output of the program.

2. Software description

EvoDyn-3s is a computable document written in Mathematica language. The document contains the executable program, detailed instructions on how to use it, and the source code. There is no need to compile the code. The program can be used directly, simply opening the computable document with the free Wolfram CDF Player or with Mathematica.

1 EvoDyn-3s’s full functionality requires Mathematica.

---

2 The payoff matrix is not updated automatically, but only when the user clicks on the button “update”. This is a purposeful implementation, since the user often wants to change several payoff values at the same time.
The next parameter in the interface is the dynamic. A pop-up menu shows the list of available evolutionary game dynamics, which are detailed in the Appendix. These constitute what we call here baseline dynamics, since they do not consider noise.

Parameter $\mu$, called “probability of random strategy”, or “noise”, modifies the baseline dynamics by considering a flow of individuals who adopt a random strategy when revising their current one. This selection of random strategies may be based on mutations, errors, or experimentation. Formally, if $\dot{x}_i = f(x_1, x_2, x_3)$ corresponds to one of the baseline evolutionary dynamics detailed in the Appendix, the evolutionary dynamics considered when $\mu \neq 0$ are given by $\dot{x}_i = (1 - \mu)f(x_1, x_2, x_3) + \mu(\frac{1}{3} - x_i)$. Small modifications of this sort can have important qualitative effects on game dynamics (see e.g. \cite{13, 14, and 15}).

The user can also easily choose the number of trajectories represented in the simplex (none, few, some or many), the length of the lines showing those trajectories, and the color gradient used to represent the speed of the dynamic.

Parameter maxTimeOut establishes the maximum time that the program is allowed to spend calculating the exact rest points of the dynamics. The default value is 1 s, which is enough in most cases.

Finally, the are two buttons which implement functionality that requires running EvoDyn-3s in Mathematica:

- The “Export figure” button generates a camera-ready vector pdf file of the phase portrait, including a table showing the components of rest points and a table showing the numerical approximations to the isolated rest points and to their corresponding eigenvalues (for dynamics where the Jacobian is defined).
- The “Generate full report” button generates a Mathematica notebook detailing the equations corresponding to the selected dynamics, the phase portrait, and the exact representations of the components of rest points, of the isolated rest points and of the eigenvalues of the Jacobian (if it is defined). The report also includes numerical approximations to the exact values.

### 3. Illustrative examples

#### 3.1. The hypercycle system

The hypercycle system is an ordinary differential equation introduced in \cite{16} to model the organization of self-replicating molecules connected in a cyclic, autocatalytic way. The equation can be represented as an instance of the replicator dynamics, so the hypercycle system for a mixture of three macromolecules can...
be directly modeled with EvoDyn-3s. Fig. 3 shows the values of the parameters required to model this system, and the output thus obtained. It is clear that all interior solutions of this hypercycle system converge to the barycenter of the simplex, as proved by [17] and [18]. EvoDyn-3s reports the exact eigenvalues of the Jacobian at the barycenter, i.e. $\frac{1}{2} \left(-1 \pm i\sqrt{3}\right)$, and their numerical approximations $-0.167 \pm 0.289i$. 

Fig. 2. Overall architecture of EvoDyn-3s. Symbol N in the figure denotes a numerical approximation.
3.2. Zeeman’s game

Zeeman’s game was introduced in [19] to show that asymptotically stable states under the replicator dynamic do not necessarily correspond to Evolutionarily Stable Strategies (ESS). Fig. 4 shows the values of the parameters required to model Zeeman’s game, and the output thus obtained. The interior Nash equilibrium of this game (i.e. the barycenter of the simplex) is not an ESS but is nevertheless asymptotically stable. This is clearly seen by noticing that the real part of the two eigenvalues of the Jacobian at this rest point is negative. EvoDyn-3s reports the exact eigenvalues of the Jacobian at the barycenter, i.e. \(-\frac{1}{3} \pm i \frac{\sqrt{3}}{3}\), and their numerical approximations \(-0.333 \pm 0.471i\).

4. Impact

The following features of EvoDyn-3s distinguish it from alternative programs such as Dynamo, PDToolbox and egplot, and make it especially useful for research:

- The selection of evolutionary dynamics considered in EvoDyn-3s (which is detailed in the Appendix) includes new evolutionary dynamics, based on interactions with partial information. Some of these dynamics are not available on any other program and are studied only in recent papers (e.g. [20,21]).

- Besides decimal approximations, EvoDyn-3s provides the rest points of the selected dynamics and the eigenvalues at isolated rest points (where the Jacobian is defined) using exact arithmetic. This implies, in particular, that if a solution is a rational number that does not admit a finite decimal representation, EvoDyn-3s provides the rational number. Similarly, if a solution is an algebraic number that does not admit a rational representation, EvoDyn-3s provides the algebraic number. Exact calculation of rest points is essential for rigorous local stability analysis [20].

- EvoDyn-3s can be run using free software. It does not require expensive software licenses, such as Mathematica or Matlab. EvoDyn-3s can be run with the free Wolfram CDF Player, which offers desktop functionality for Windows, Mac and Linux systems, and it also runs on some mobile platforms (e.g. those running iOS). The only functionality missing when running EvoDyn-3s on free software is the dynamic creation of output files.

- EvoDyn-3s has been designed to be used through a simple and intuitive user interface. It does not require any programming knowledge whatsoever. The user only needs to

---

3 Beyond degree four, most polynomials do not have roots that can be expressed in terms of radicals. An example of an algebraic number that cannot be expressed in terms of radicals is any of the roots of the polynomial \(x^5 - x - 1\).
specify the payoff matrix of the game and choose one of the available dynamics using the mouse.

To provide an idea of the interest and potential applications of this type of software, we note that –even with the barrier entry caused by the need to own a Mathematica license and possess some (basic) knowledge of the Mathematica environment – Dynamo has been used as supporting software for cutting-edge research and for generating high-quality graphs in articles published in leading journals in the field, such as Games and Economic Behavior [22,23], Journal of Economic Theory [24], Journal of Theoretical Biology [25], Theoretical Economics [26], Proceedings of the Royal Society of London [27] and Proceedings of the National Academy of Sciences [28].

5. Conclusions

EvoDyn-3s is open-source software that can generate phase portraits of various evolutionary game dynamics and data to analyze their equilibria. It includes classic and new evolutionary dynamics, it can run on free software and on different platforms, it does not require any programming knowledge and it provides exact results for rest points and their corresponding eigenvalues. Besides being useful to analyze evolutionary dynamics, it is intended to generate camera-ready high-quality graphs for publication.

Acknowledgments

Financial support from NSF Grant SES-1728853, U.S. Army Research Office Grant MSN201957, grants PRX15/00362 and PRX16/00048 awarded by the Spanish MECD, and Spanish Ministry of Science and Innovation’s project ECO2017-83147-C2-2-P (MINECO/AEI/FEDER, UE) is gratefully acknowledged. We are also very grateful to the community of https://mathematica.stackexchange.com.

Appendix. Evolutionary game dynamics formulas

The baseline evolutionary dynamics implemented in EvoDyn-3s are:

- Replicator [1–4]. The growth rate of each strategy share is proportional to its prevalence $x_i$ and to the difference ($\pi_i - \overline{\pi}$) between the expected payoff of that strategy $\pi_i = \sum_{j=1}^{3} a_{ij} x_j$ and the average expected payoff in the population $\overline{\pi} = \sum_{i=1}^{3} x_i \pi_i$.
  
  \[
  \dot{x}_i = x_i (\pi_i - \overline{\pi})
  \]

- Smith [29,30]. The flow from strategy $j$ to strategy $i$ is proportional to the prevalence $x_j$ of strategy $j$ and to the
expected payoff difference $\pi_i - \pi_j$, as long as this difference is positive.

\[
\dot{x}_i = \sum_{j=1}^{3} x_j [\pi_i - \pi_j]_+ - x_i \sum_{j=1}^{3} [\pi_j - \pi_i]_+ + \lambda
\]

where the function $[a]_+$ is such that $[a]_+ = a$ if $a > 0$, and $[a]_+ = 0$ if $a \leq 0$.

- **Brown–von Neumann–Nash** [31–36]. The flow from strategy $j$ to strategy $i$ is proportional to the difference $\pi_j - \pi_i$, and to the expected payoff difference $\pi_i - \pi_j$, if it is positive.

\[
\dot{x}_i = [\pi_i - \pi_j]_+ - x_i \sum_{j=1}^{3} [\pi_j - \pi_i]_+
\]

- **Logit** [37–39]. The flow from strategy $j$ to strategy $i$ is proportional to the prevalence $x_j$ of strategy $j$ and to the expected payoff difference $\pi_i - \pi_j$, if it is positive.

\[
\dot{x}_i = e^{\eta x_i - 1} \sum_{j=1}^{3} e^{\eta x_j - 1} - x_i
\]

Parameter $\eta$ modulates the impact of payoff differences on the dynamics. As $\eta \to 0$ the impact of payoff differences gets more acute and the Logit dynamics approach the best response dynamics.

- **Single-match imitative logit.** This is the mean dynamics of an imitation process in which players obtain a payoff in a random match with another player and occasionally revise their strategy. A revising agent looks at another randomly chosen player (and his obtained payoff) and chooses a strategy with probability proportional to the logit factors of the two obtained payoffs.

\[
\dot{x}_i = x_i \sum_{j=1}^{3} \sum_{k=1}^{3} x_{jk} \sum_{m=1}^{3} \frac{e^{\eta x_m - 1}}{e^{\eta x_i - 1}} - \frac{e^{\eta x_j - 1}}{e^{\eta x_i - 1}}
\]

- **Imitate if better in one match** [40]. This is the mean dynamics of an imitation process in which players obtain a payoff in a random match with another player and occasionally revise their strategy. A revising agent looks at another randomly chosen player. If the payoff obtained by the other player is higher, the strategy of the other player is adopted.

\[
\dot{x}_i = x_i \sum_{j=1}^{3} \sum_{k=1}^{3} x_{jk} \sum_{m=1}^{3} \text{sign}(x_i - x_m)
\]

- **Test another in one match** [20,41]. This is the mean dynamics of an adaptive process in which players obtain a payoff in a random match with another player and occasionally revise their strategy. A revising agent tests another alternative random strategy in a new random match. If the outcome obtained with the alternative strategy is preferred, that strategy is adopted.

\[
\dot{x}_i = \frac{1}{3} \sum_{j=1}^{3} \sum_{k=1}^{3} x_{jk} \sum_{m=1}^{3} \text{sign}(x_i - x_m)
\]

where the “greater than” function $I_{a>b}$ is defined by $I_{a>b} = 1$ if $a > b$ and $I_{a>b} = 0$ if $a \leq b$.

- **Best response to one random player** [21,42,43]. This is the mean dynamics of an adaptive process in which players occasionally revise their strategy. A revising agent looks at another randomly chosen player and adopts a best response to the strategy of that random player.

\[
\dot{x}_i = \left( \sum_{j \in BR(i)} x_j \right) - x_i
\]

where $BR(j)$ is the subset of strategies that are a best response to strategy $j$, i.e., $i : a_i = \max_i (a_{ij})$.

References


[34] Swinkels JM. Adjustment dynamics and rational play in games. Games Econ Behav 1993;5:455–84.


