Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

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.

Code metadata

Current code version	v1.0
Permanent link 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
If available Link to developer documentation/manual	Documentation included within the computable document
Support email for questions	lrizquierdo@ubu.es

Software metadata

Current software version	v1.0
Permanent link 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*.
If available, link to user manual - if formally published include a reference to the publication in the reference list	Documentation included within the computable document
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).
URLs: http://www.luis.izqui.org (L.R. Izquierdo), http://www.segis.izqui.org (S.S. Izquierdo), http://www.ssc.wisc.edu/~whs (W.H. Sandholm).

https://doi.org/10.1016/j.softx.2018.07.006

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 systems 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 when analyzing adaptive processes.

In general, many of the basic properties of evolutionary game dynamics can be illustrated using games with three strategies [5,6], and applications to the evolution of cooperation are often studied in this framework [7]. To understand and analyze evolutionary game dynamics with three strategies, one of the most useful and intuitive tools are phase portraits, which are geometric representations of the trajectories of the dynamical system in the phase plane. Generating this type of graph typically requires expensive software or customized programming, and this can present a considerable barrier for many researchers studying evolutionary processes. *EvoDyn-3s* has been designed to overcome this barrier.

Specifically, *EvoDyn-3s* provides high-quality print-ready vector phase portraits for a diverse group of evolutionary game dynamics with three strategies. Using it does not require any programming and most of its functionality can be run with the free *Wolfram CDF Player*.[1] *EvoDyn-3s* also calculates and presents the equilibria of the selected dynamics and performs an eigenvalue analysis of the linearized dynamics *using exact arithmetic*, a feature that is not available in other programs and can be very useful for theoretical analysis.

The software most closely related to *EvoDyn-3s* is *Dynamo* [8,9]. *Dynamo* is also open-source software that runs on *Mathematica*, and has also been designed to create phase diagrams and other images related to dynamical systems from evolutionary game theory. *Dynamo* can be used to generate graphs for single-population games with 3 or 4 strategies and for some multipopulation games. *Dynamo* is more flexible and general than *EvoDyn-3s*, but significantly less user-friendly, and it uses numerical approximations rather than exact arithmetic. Another software somewhat related to *EvoDyn-3s* is *PDToolbox* [10], which is a set of functions coded in *Matlab* for analyzing some evolutionary dynamics, as well as finite-population agent-based models related to those dynamics. The Python package *egtplot* [11] creates phase diagrams for the replicator dynamics. While narrower in scope and less user-friendly than *EvoDyn-3s*, *egtplot* has the commendable feature of running on an open-source platform. Lastly, *ABED* [12] (**A**gent-**B**ased **E**volutionary **D**ynamics) is also free and completely open-source software for simulating adaptive processes, but in finite populations. It provides a complementary approach to the analysis of evolutionary dynamics followed in *EvoDyn-3s*, in the sense that many of the adaptive processes considered and implemented in *ABED* (following an agent-based approach) can be approximated –for sufficiently large populations – by differential equations corresponding to the evolutionary dynamics implemented in *EvoDyn-3s*.

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, by simply opening the computable document with the free *Wolfram CDF Player* or with *Mathematica*.

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.[2] Thus, for example, the user can gradually move any parameter slider and immediately appreciate how this change affects the output of the program.

2.1. Software architecture

EvoDyn-3s conducts the following high-level operations, which are sketched in Fig. 2:

- Creates the system of differential equations using the following input provided by the user: payoff matrix, baseline dynamic, probability of random strategy μ, and –only for dynamics Logit and Single-match imitative logit – parameter η. The generated system of differential equations is of the form $(\dot{x}_1, \dot{x}_2, \dot{x}_3) = f(x_1, x_2, x_3)$, where x_i represents the fraction of the population using strategy i.
- Solves the system of differential equations numerically for various initial conditions and represents the solutions. The number of initial conditions and the length of the computed trajectories can be controlled by the user.
- Represents a series of orbits and arrows showing the direction of movement in the 2-dimensional simplex $\{(x_1, x_2, x_3) \mid \sum_{i=1}^{3} x_i = 1 \text{ and } x_i \geq 0\}$.
- Colors the background of the simplex according to the speed of the dynamic, using the color gradient selected by the user.
- Computes the rest points of the dynamic using exact arithmetic.
- Computes numerical approximations to the isolated rest points. Shows them in a table and represents them in the 2-dimensional simplex.
- For dynamics where the Jacobian is defined, computes the eigenvalues of the Jacobian of the dynamic at the exact isolated rest points using exact arithmetic. Computes numerical approximations to the eigenvalues and shows them in a table.
- Shows the components of rest points in a table and represents them in the 2-dimensional simplex.

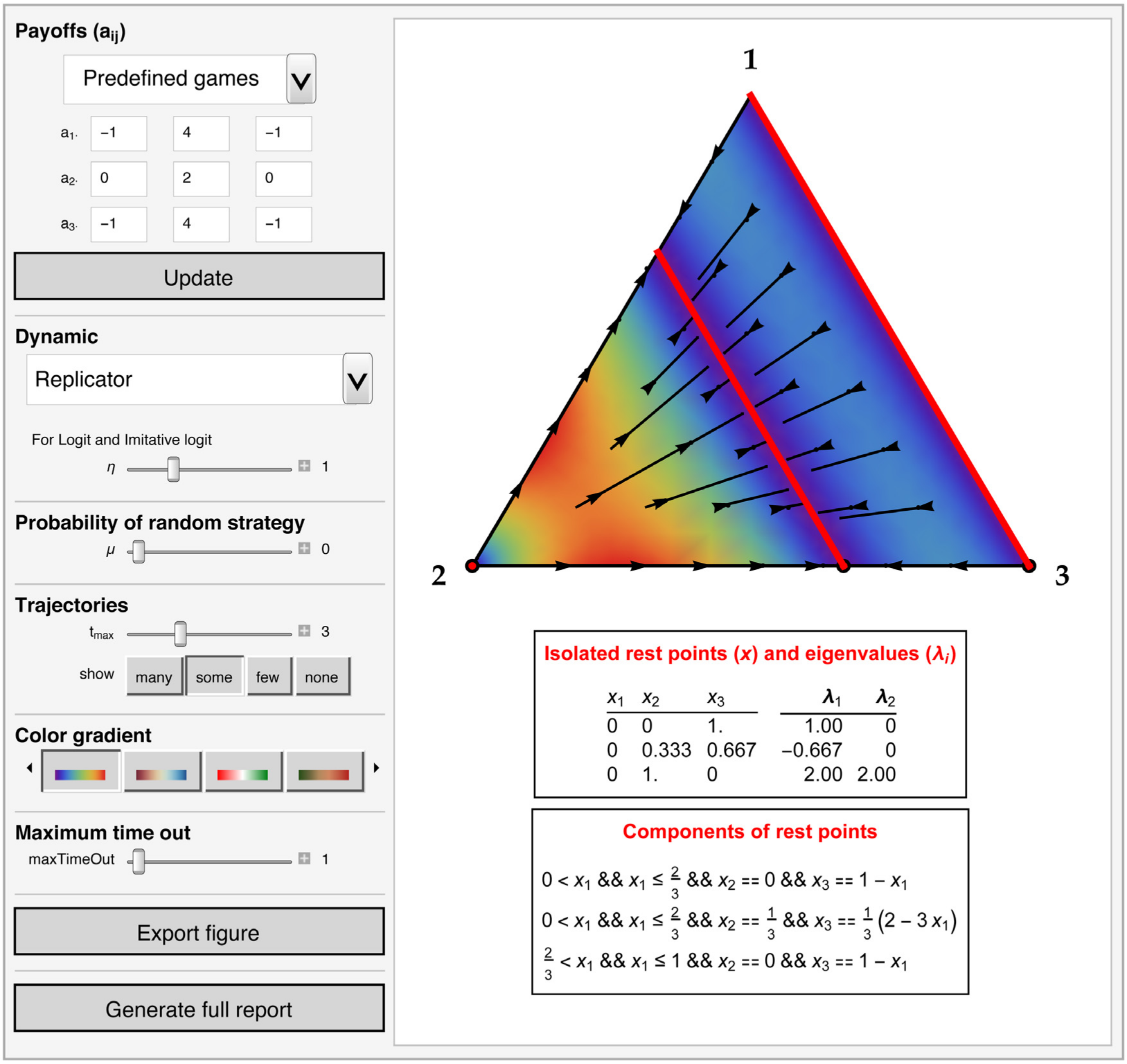
2.2. Software functionalities

EvoDyn-3s generates phase portraits of evolutionary game dynamics, colors 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 × 3 game by setting the values a_{ij} of the 3 × 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.

[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.



Payoffs (a_{ij})

$a_{1\cdot}$	−1	4	−1
$a_{2\cdot}$	0	2	0
$a_{3\cdot}$	−1	4	−1

Dynamic: Replicator

η 1

μ 0

t_{max} 3

maxTimeOut 1

Isolated rest points (x) and eigenvalues (λ_i)

x_1	x_2	x_3	λ_1	λ_2
0	0	1.	1.00	0
0	0.333	0.667	−0.667	0
0	1.	0	2.00	2.00

Components of rest points

$0 < x_1$ && $x_1 \leq \frac{2}{3}$ && $x_2 == 0$ && $x_3 == 1 - x_1$

$0 < x_1$ && $x_1 \leq \frac{2}{3}$ && $x_2 == \frac{1}{3}$ && $x_3 == \frac{1}{3}(2 - 3\,x_1)$

$\frac{2}{3} < x_1$ && $x_1 \leq 1$ && $x_2 == 0$ && $x_3 == 1 - x_1$

Fig. 1. Interface of *EvoDyn-3s*.

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 μ, 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 $\dot{x}_i = f_i(x_1, x_2, x_3)$ corresponds to one of the baseline evolutionary dynamics detailed in the Appendix, the evolutionary dynamics considered when $\mu \neq 0$ are given by $\dot{x}_i = (1-\mu)f_i(x_1, x_2, x_3) + \mu(\frac{1}{3} - x_i)$. Small modifications of this sort can have important qualitative effects on game dynamics (see e.g. [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.

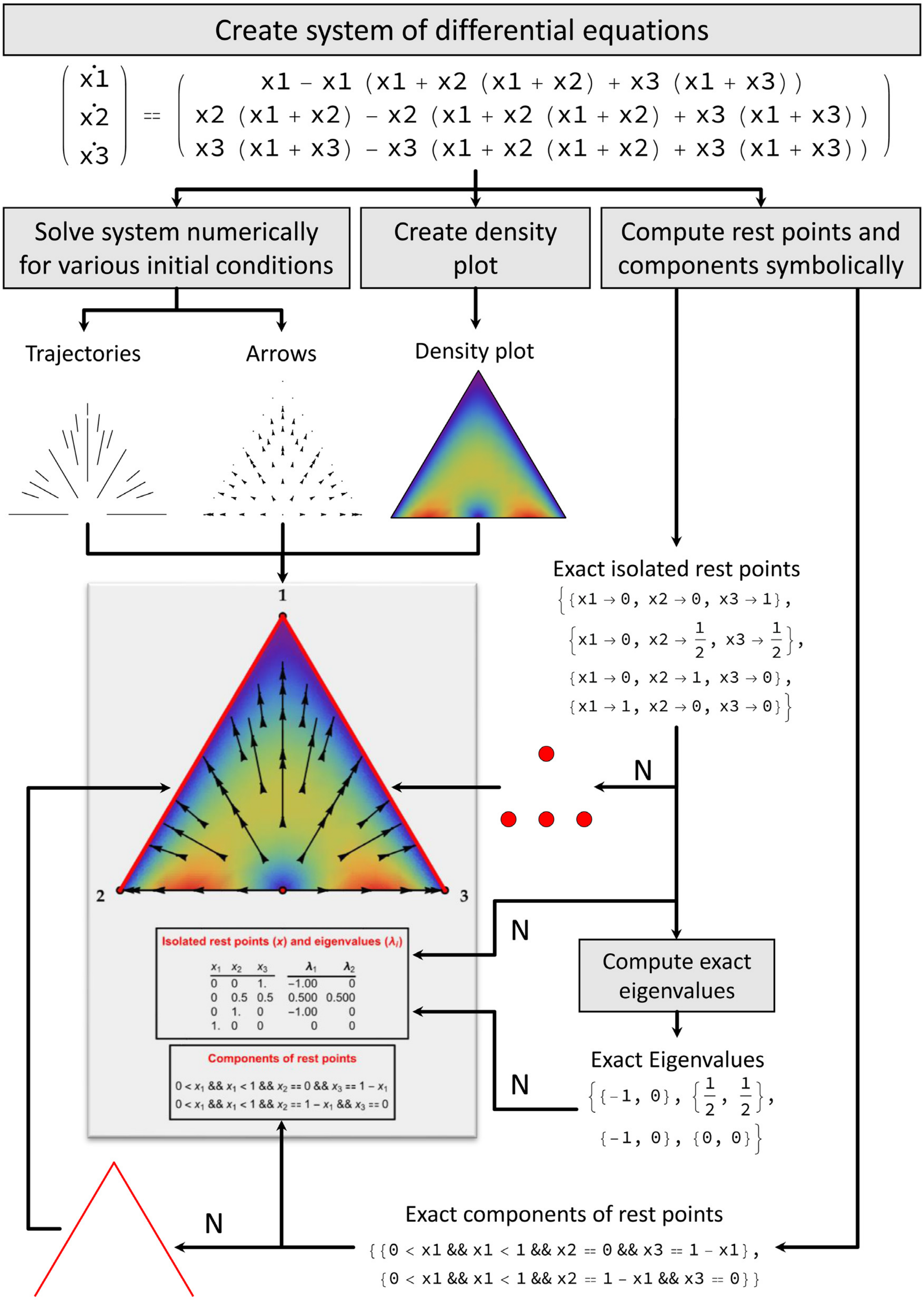
Finally, the are two buttons which implement functionality that requires running *EvoDyn-3s* in *Mathematica*:

- 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 [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


Create system of differential equations
x1 − x1 (x1 + x2 (x1 + x2) + x3 (x1 + x3))
x2 (x1 + x2) − x2 (x1 + x2 (x1 + x2) + x3 (x1 + x3))
x3 (x1 + x3) − x3 (x1 + x2 (x1 + x2) + x3 (x1 + x3))
Solve system numerically for various initial conditions
Create density plot
Compute rest points and components symbolically
Trajectories
Arrows
Density plot
1
2
3
Exact isolated rest points
{{x1 → 0, x2 → 0, x3 → 1}, {x1 → 0, x2 → 1/2, x3 → 1/2}, {x1 → 0, x2 → 1, x3 → 0}, {x1 → 1, x2 → 0, x3 → 0}}
N
N
Compute exact eigenvalues
Exact Eigenvalues
{{−1, 0}, {1/2, 1/2}, {−1, 0}, {0, 0}}
N
N
Exact components of rest points
{{0 < x1 && x1 < 1 && x2 == 0 && x3 == 1 − x1}, {0 < x1 && x1 < 1 && x2 == 1 − x1 && x3 == 0}}

Isolated rest points (x) and eigenvalues (λ_i)

x_1	x_2	x_3	λ_1	λ_2
0	0	1.	−1.00	0
0	0.5	0.5	0.500	0.500
0	1.	0	−1.00	0
1.	0	0	0	0

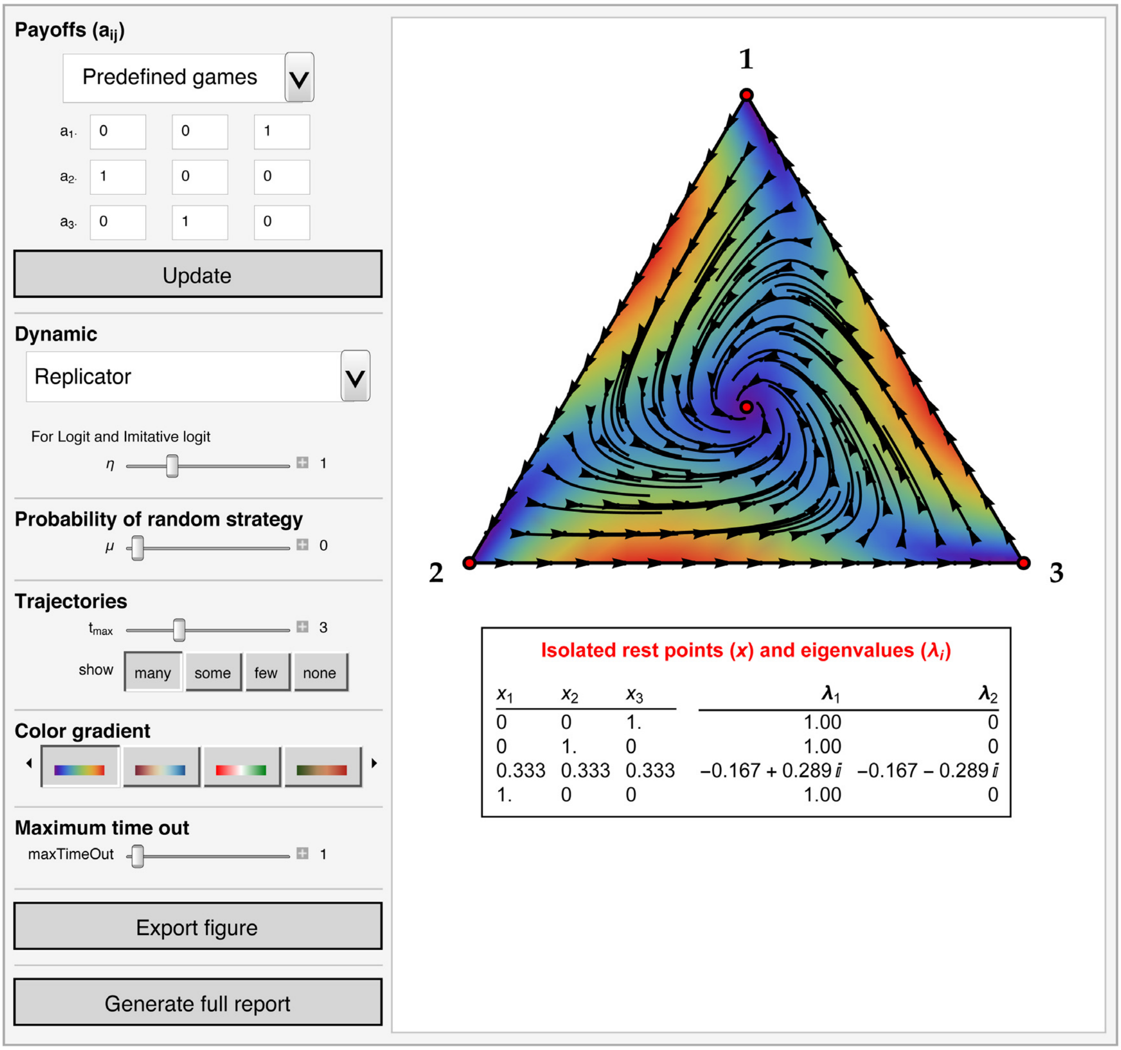
Components of rest points

$0 < x_1 \ \&\& \ x_1 < 1 \ \&\& \ x_2 == 0 \ \&\& \ x_3 == 1 - x_1$

$0 < x_1 \ \&\& \ x_1 < 1 \ \&\& \ x_2 == 1 - x_1 \ \&\& \ x_3 == 0$

Fig. 2. Overall architecture of *EvoDyn-3s*. Symbol N in the figure denotes a numerical approximation.

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. $\frac{1}{6}\left(-1 \pm i\sqrt{3}\right)$, and their numerical approximations $-0.167 \pm 0.289i$.



Payoffs (a_{ij})

Predefined games

$a_{1\cdot}$	0	0	1
$a_{2\cdot}$	1	0	0
$a_{3\cdot}$	0	1	0

Dynamic

Replicator

For Logit and Imitative logit

η 1

Probability of random strategy

μ 0

Trajectories

t_{max} 3

show many some few none

Color gradient

Maximum time out

maxTimeOut 1

Isolated rest points (x) and eigenvalues (λ_i)

x_1	x_2	x_3	λ_1	λ_2
0	0	1.	1.00	0
0	1.	0	1.00	0
0.333	0.333	0.333	−0.167 + 0.289 i	−0.167 − 0.289 i
1.	0	0	1.00	0

Fig. 3. The hypercycle system with three macromolecules.

3.2. *Zeeman's game*

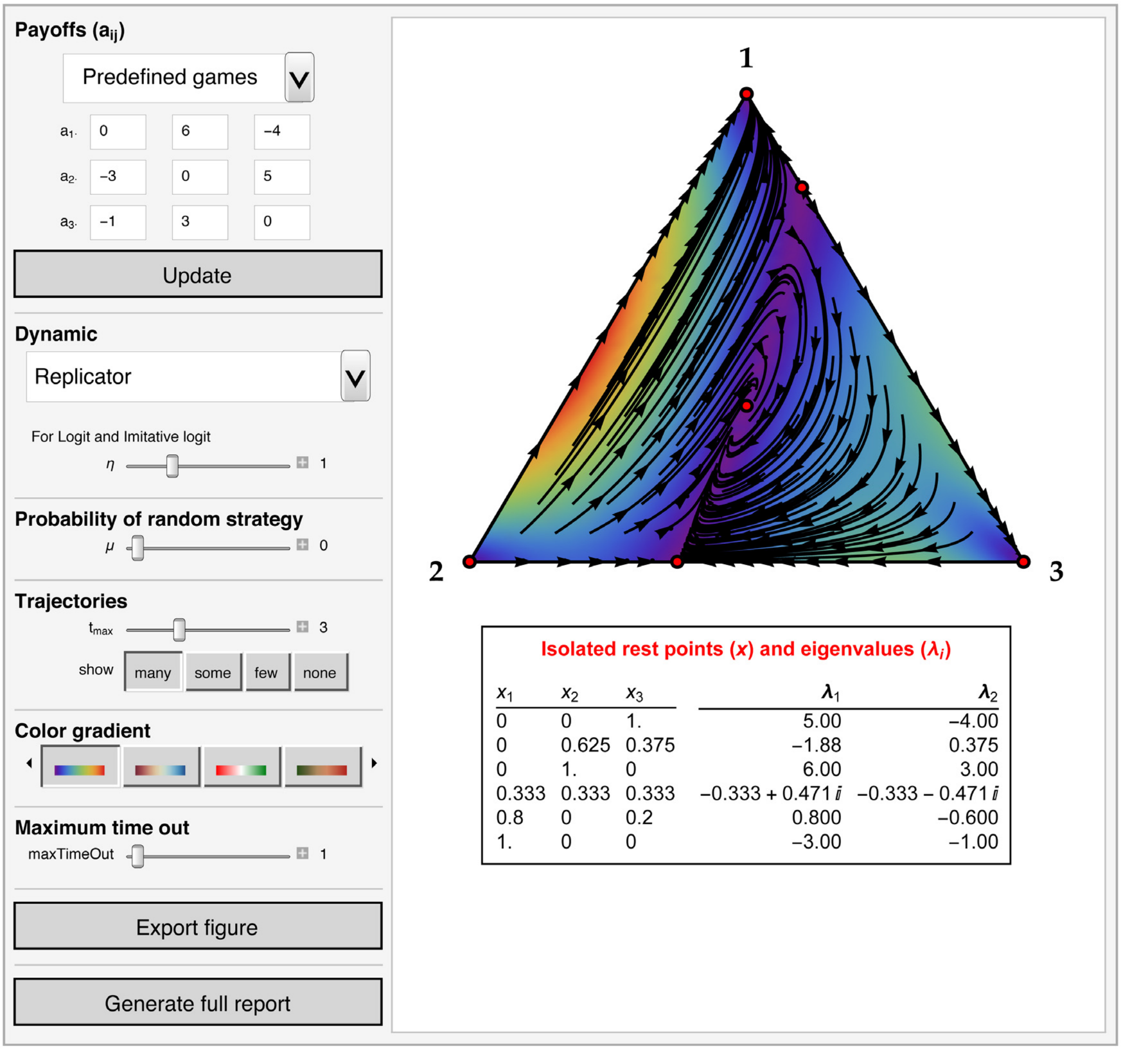
Zeeman's game was introduced in [19] to show that asymptotically stable states under the replicator dynamic do not necessarily correspond to Evolutionarily Stable Strategies (ESS). Fig. 4 shows the values of the parameters required to model Zeeman's game, and the output thus obtained. The interior Nash 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. $-\frac{1}{3} \pm i\frac{\sqrt{2}}{3}$, and their numerical approximations $-0.333 \pm 0.471i$.

4. Impact

The following features of *EvoDyn-3s* distinguish it from alternate 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.[3] 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 $x^5 - x - 1$.



Payoffs (a_{ij})

$a_{1\cdot}$	0	6	−4
$a_{2\cdot}$	−3	0	5
$a_{3\cdot}$	−1	3	0

Dynamic: Replicator

For Logit and Imitative logit

η 1

Probability of random strategy

μ 0

Trajectories

t_{max} 3

Color gradient

Maximum time out

maxTimeOut 1

Isolated rest points (x) and eigenvalues (λ_i)

x_1	x_2	x_3	λ_1	λ_2
0	0	1.	5.00	−4.00
0	0.625	0.375	−1.88	0.375
0	1.	0	6.00	3.00
0.333	0.333	0.333	−0.333 + 0.471 i	−0.333 − 0.471 i
0.8	0	0.2	0.800	−0.600
1.	0	0	−3.00	−1.00

Fig. 4. The replicator dynamics in Zeeman's game.

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 $(\pi_i - \bar{\pi})$ 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 $\bar{\pi} = \sum_{i=1}^{3} x_i\pi_i$.

  $$\dot{x}_i = x_i(\pi_i - \bar{\pi})$$

- 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.

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

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 prevalence x_j of strategy j and to the expected payoff difference $\pi_i - \bar{\pi}$, if it is positive.

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

- Logit [37–39]. The flow from strategy j to strategy i is proportional to the prevalence x_j of strategy j and to the logit term $e^{\pi_i \eta^{-1}}(\sum_k e^{\pi_k \eta^{-1}})^{-1}$.

$$\dot{x}_i = \frac{e^{\pi_i \eta^{-1}}}{\sum_k e^{\pi_k \eta^{-1}}} - x_i$$

Parameter η modulates the impact of payoff differences on the dynamics. As $\eta \rightarrow 0$ 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.

$$\dot{x}_i = x_i \sum_{j=1, j\neq i}^{3} \sum_{k=1}^{3} \sum_{m=1}^{3} x_j\, x_k\, x_m \frac{e^{a_{ik}\eta^{-1}} - e^{a_{jm}\eta^{-1}}}{e^{a_{ik}\eta^{-1}} + e^{a_{jm}\eta^{-1}}}$$

- 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.

$$\dot{x}_i = x_i \sum_{j=1}^{3} \sum_{k=1}^{3} \sum_{m=1}^{3} x_j\, x_k\, x_m \,\mathrm{sign}(a_{ik} - a_{jm})$$

- 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.

$$\dot{x}_i = \frac{1}{2} \sum_{j=1, j\neq i}^{3} \sum_{k=1}^{3} \sum_{m=1}^{3} [x_k\, x_m\, (x_j\, I_{\{a_{ik} > a_{jm}\}} - x_i\, I_{\{a_{jm} > a_{ik}\}})]$$

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.

$$\dot{x}_i = \left(\sum_{j:i\in BR(j)} \frac{x_j}{\#BR(j)} \right) - x_i$$

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

References

[1] Taylor PD, Jonker L. Evolutionarily stable strategies and game dynamics. Math Biosci 1978;40:145–56.

[2] Helbing D. A mathematical model for behavioral changes by pair interactions. In: Haag G, Mueller U, Troitzsch KG, editors. Economic evolution and demographic change: formal models in social sciences. Berlin: Springer; 1992. p. 330–48.

[3] Schlag KH. Why imitate, and if so, how? a boundedly rational approach to multi-armed bandits. J Econom Theory 1998;78:130–56.

[4] Hofbauer J. Imitation dynamics for games, Unpublished manuscript, 1995, University of Vienna.

[5] Weibull JW. Evolutionary game theory. Cambridge: MIT Press; 1995.

[6] Sandholm WH. Population games and evolutionary dynamics. Cambridge: MIT Press; 2010.

[7] Sigmund K. The calculus of selfishness. Princeton: Princeton University Press; 2010.

[8] Sandholm WH, Dokumacı E, Franchetti F. Dynamo: diagrams for evolutionary game dynamics, Software. http://www.ssc.wisc.edu/~whs/dynamo.

[9] Franchetti F, Sandholm WH. An introduction to *dynamo*: diagrams for evolutionary game dynamics. Biol Theory 2013;8(2):167–78. http://dx.doi.org/10.1007/s13752-013-0109-z.

[10] Barreto C. Population dynamics Toolbox (PDToolbox) Software. https://github.com/carlobar/PDToolbox_matlab.

[11] Mirzaev I, Williamson DFK, Scott JG. egtplot: A python package for three-strategy evolutionary games. J. Open Source Softw. 2018;3(26):735. http://dx.doi.org/10.21105/joss.00735.

[12] Izquierdo LR, Izquierdo SS, Sandholm WH. An introduction to ABED: agent-based simulation of evolutionary game dynamics, Unpublished manuscript, 2018.

[13] Bomze IM, Burger R. Stability by mutation in evolutionary games. Games Econom Behav 1995;11(2):146–72. http://dx.doi.org/10.1006/game.1995.1047.

[14] Binmore K, Samuelson L. Evolutionary drift and equilibrium selection. Rev Econom Stud 1999;66:363–93.

[15] Imhof LA, Fudenberg D, Nowak MA. Evolutionary cycles of cooperation and defection. Proc Natl Acad Sci 2005;31:10797–800. http://dx.doi.org/10.1073/pnas.0502589102.

[16] Eigen M, Schuster P. The hypercycle: a principle of natural self-organization. Berlin: Springer; 1979.

[17] Schuster P, Sigmund K, Wolff R. Dynamical systems under constant organization i: Topological Analysis of a Family of Nonlinear Differential Equations—A Model for Catalytic Hypercycles. Bull Math Biol 1978;40:743–69.

[18] Hofbauer J, Schuster P, Sigmund K. Competition and cooperation in catalytic self-replication. J Math Biol 1981;11:155–68.

[19] Zeeman EC. Population dynamics from game theory. In: Nitecki Z, Robinson C, editors. Global theory of dynamical systems (Evanston, 1979). Lecture notes in mathematics, vol. 819, Berlin: Springer; 1980. p. 472–97.

[20] Sandholm WH, Izquierdo SS, Izquierdo LR. Best experienced payoff dynamics and cooperation in the centipede game, Unpublished manuscript, 2017.

[21] Oyama D, Sandholm WH, Tercieux O. Sampling best response dynamics and deterministic equilibrium selection. Theoret. Econ. 2015;10:243–81.

[22] Mohlin E. Evolution of theories of mind. Games Econom Behav 2012;75:299–318.

[23] van Veelen M. Robustness against indirect invasions. Games Econom Behav 2012;74:382–93.

[24] Hofbauer J, Sandholm WH. Stable games and their dynamics. J Econom Theory 2009;144:1665–93.

[25] Arenas A, Camacho J, Cuesta JA, Requejo RJ. The joker effect: cooperation driven by destructive agents. J Theoret Biol 2011;279(1):113–9. http://dx.doi.org/10.1016/j.jtbi.2011.03.017.

[26] Hofbauer J, Sandholm WH. Survival of dominated strategies under evolutionary dynamics. Theoret Econ 2011;6:341–77.

[27] McNally L, Jackson AL. Cooperation creates selection for tactical deception. Proc R Soc Lond Ser B Biol Sci 2013;280(1762):. http://dx.doi.org/10.1098/rspb.2013.0699.

[28] Traulsen A, Hauert C, De Silva H, Nowak MA, Sigmund K. Exploration dynamics in evolutionary games. Proc Natl Acad Sci 2009;106(3):709–12. http://dx.doi.org/10.1073/pnas.0808450106.

[29] Smith MJ. The stability of a dynamic model of traffic assignment—an application of a method of Lyapunov. Transp Sci 1984;18:245–52.

[30] Sandholm WH. Pairwise comparison dynamics and evolutionary foundations for nash equilibrium. Games 2010;1:3–17.

[31] Brown GW, von Neumann J. Solutions of games by differential equations. In: Kuhn HW, Tucker AW, editors. Contributions to the theory of games I. Annals of mathematics studies, vol. 24, Princeton: Princeton University Press; 1950. p. 73–9.

[32] Nash JF. Non-Cooperative games. Ann of Math 1951;54:287–95.

[33] Skyrms B. The dynamics of rational deliberation. Cambridge: Harvard University Press; 1990.

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

[35] Weibull JW. The mass action interpretation. excerpt from the work of John Nash in game theory: Nobel Seminar, December 8, 1994. J Econom Theory 1996;69:165–71.

[36] Hofbauer J. From Nash and Brown to Maynard Smith: equilibria, dynamics, and ESS. Selection 2000;1:81–8.

[37] Blume LE. Population games. In: Arthur WB, Durlauf SN, Lane DA, editors. The economy as an evolving complex system II. Reading, MA: Addison-Wesley; 1997. p. 425–60.

[38] Fudenberg D, Levine DK. The theory of learning in games. Cambridge: MIT Press; 1998.

[39] Hofbauer J, Sandholm WH. Evolution in games with randomly disturbed payoffs. J Econom Theory 2007;132:47–69.

[40] Izquierdo SS, Izquierdo LR. Stochastic approximation to understand simple simulation models. J Stat Phys 2013;151(1):254–76. http://dx.doi.org/10.1007/s10955-012-0654-z.

[41] Sethi R. Stability of equilibria in games with procedurally rational players. Games Econom Behav 2000;32:85–104.

[42] Sandholm WH. Almost global convergence to *p*-dominant equilibrium. Internat J Game Theory 2001;30:107–16.

[43] Kosfeld M, Droste E, Voorneveld M. A myopic adjustment process leading to best reply matching. J Econom Theory 2002;40:270–98.