E&F Chaos: A User Friendly Software Package for Nonlinear Economic Dynamics

Cees Diks · Cars Hommes · Valentyn Panchenko · Roy van der Weide

Abstract   The use of nonlinear dynamic models in economics and finance has expanded rapidly in the last two decades. Numerical simulation is crucial in the investigation of nonlinear systems. *E&F Chaos* is an easy-to-use and freely available software package for simulation of nonlinear dynamic models to investigate stability of steady states and the presence of periodic orbits and chaos by standard numerical simulation techniques such as time series, phase plots, bifurcation diagrams, Lyapunov exponent plots, basin boundary plots and graphical analysis. The package contains many well-known nonlinear models, including applications in economics and finance, and is easy to use for non-specialists. New models and extensions or variations are easy to implement within the software package without the use of a compiler or other software. The software is demonstrated by investigating the dynamical behavior of some simple examples of the familiar cobweb model, including an extension with heterogeneous agents and asynchronous updating of strategies. Simulations with the *E&F Chaos* software quickly provide information about local and global dynamics and easily lead to challenging questions for further mathematical analysis.

C. Diks (✉) · C. Hommes
CeNDEF, Department of Quantitative Economics, University of Amsterdam,
Roetersstraat 11, 1018 WB Amsterdam, the Netherlands
e-mail: c.g.h.diks@uva.nl

C. Hommes
e-mail: c.h.hommes@uva.nl

V. Panchenko
School of Economics, University of New South Wales, Sydney, NSW, 2052, Australia
e-mail: v.panchenko@unsw.edu.au

R. van der Weide
The World Bank, Washington, DC, USA
e-mail: rvanderweide@worldbank.org

1 Introduction

In the last two decades the theory of nonlinear dynamical systems has flourished, and a wide range of mathematical tools for the analysis of nonlinear differential equations as well as nonlinear maps have become available, see e.g. textbook treatments such as Guckenheimer and Holmes (1983), Grandmont (1988), Arrowsmith and Place (1995), Mira et al. (1996), and Medio and Lines (2001). Nonlinear dynamics tools have been widely applied in economics and finance, among others, through the work of Grandmont (1985), Boldrin and Woodford (1990), Brock et al. (1991), Hommes (1991), Medio (1992), Day (1994) and Rosser (2000). More recently, the bounded rationality and interacting agents approach in economics and finance provides a natural framework to modeling markets as nonlinear, adaptive systems, see e.g. the surveys of Hommes (2006) and LeBaron (2006). Since nonlinear dynamical systems are difficult to solve analytically, numerical analysis and simulations play a key role in the analysis of nonlinear systems and its applications.

The purpose of this paper is to discuss the main features of the *E&F Chaos* software package with a menu-driven interface for simulation of nonlinear dynamical systems. "*E&F*" stands for Economics and Finance, because the package contains a list of benchmark applications in economics and finance. Many researchers and specialists in the field have developed their own software for advanced numerical analysis of nonlinear systems. But specialized software is not always easy to use. *E&F Chaos* is easy to use since it has been developed for use by non-specialists, students and researchers in economics and finance who would like a quick start in applying standard simulation tools for nonlinear dynamic models. The software program and source code are freely available at http://www.feb.uva.nl/cendef and the numerical accuracy and quality of graphics allow direct generation of publication quality output for teaching and research. A convenient feature of the package is that new models can easily be included as a text file, without the need of a compiler or additional software. Once a new model has been included, basic tools such as time series, phase plots, bifurcation diagrams and largest Lyapunov exponents plots become immediately available. Simulation with *E&F Chaos* easily leads to challenging mathematical and economic questions for further analysis.

The paper is organized as follows. Section 2 gives some background information about the development of the *E&F Chaos* program and briefly discusses some related software packages. The most important features of *E&F Chaos* are illustrated through simulation of economic examples in Sects. 3 and 4. Although the software does include a number of nonlinear differential equation models, we restrict our attention here to nonlinear difference equations, since most nonlinear applications in economics are in discrete time. Section 3 considers an example of a one-dimensional (1-D) model, the cobweb model with adaptive expectations, whereas Sect. 4 considers an example of a higher dimensional (4-D) model, the cobweb

model with heterogeneous expectations. This section considers an extension of the cobweb model with heterogeneous expectation rules of Brock and Hommes (1997), to allow for *asynchronous updating* of strategies. The *E&F Chaos* software is used to investigate how asynchronous updating affects the dynamical behavior. Both Sects. 3 and 4 are subdivided into subsections illustrating the key features, in particular the options from the Plot menu, of the software. The paper ends with a short concluding section.

2 About E&F Chaos and Related Software

The earliest version of the *E&F Chaos* software dates back to the early nineties, when a DOS version programmed in Turbo Pascal was used in teaching of courses in dynamical systems and nonlinear economic dynamics, both at the advanced bachelor and the masters level, at the economics department of the University of Amsterdam. At the end of the nineties, a major revision of the software took place and a Windows version with a graphical user interface was developed. Both the user interface and the core routines for all calculations were programmed in Delphi 4.0.[1] This version has been around essentially unmodified since 1998, and was used in teaching the aforementioned courses as well as NAKE Ph.D courses on nonlinear economic dynamics attended by Ph.D students from all Dutch faculties of economics. Originally all (economic) models were hard-coded in Delphi Pascal. The advantage of using a low-level programming language was the speed of the resulting executable code, which could easily exceed that of more flexible interpreted languages (Basic, Mathematica) by a factor of 10–100. This was important for computationally intensive algorithms, such as creating bifurcation diagrams or determining basin boundaries of attractors. However, a clear disadvantage was that for every new model the source code had to be extended after which the program needed to be re-compiled. In practice this meant that it was not straightforward for students to analyze a new model independently. They either needed to wait for implementation of the particular model by the developers, or develop and compile parts of new code themselves. As a result, the old version was rather inflexible, and the number of 'hard-wired' models appearing in the program's menu started to grow, apparently beyond any bounds.

In this paper we give an overview of the functionality and implementation of the latest version of the *E&F Chaos* package as of June 2007. With the advent of techniques such as just-in-time compilation, scripting languages became available with the flexibility of low-level code such as Pascal or C, while their speed is comparable to that of compiled code. This allows one to specify the model code in such a scripting language, and skip the cumbersome compilation step, practically without any loss of speed. Since each model can then be specified in a text file that can be interpreted 'on the fly' one may simply work with a model by loading it, so that the long menus can be dispensed of easily. We have chosen for the open source scripting language LUA as

---

[1] We are indebted to Remco Peters, who, together with Roy van der Weide, programmed the initial Delphi code.

it allows high flexibility as well as speed. A second recent addition is the replacement of bitmap graphics by high-quality Encapsulated Postscript (EPS) figures. To this end we use the *gnuplot* program (see Racine 2006). To prevent the user from having to download and install *gnuplot* separately, and to avoid possible incompatibilities with future versions of *gnuplot*, we have chosen to distribute the necessary *gnuplot* files with *E&F Chaos*.

2.1 Comparison with Existing Software

Many packages are available for the simulation and analysis of dynamical systems. Since it is beyond the scope of this paper to review all of them, we refer the interested reader to the dynamical systems web portal at http://www.dynamicalsystems. org/sw/sw/, where users can choose their preferred package that best matches their computational needs. One can distinguish many different types of simulation software. Several dynamical systems software packages are implemented as toolboxes for multi-purpose high-level programming environments such as Matlab, Octave, S-Plus or R. These environments are suitable for simulating a single or a few time series and generating high-level graphical output, for instance of phase plots of the attractors. For more computationally intensive graphs, such as bifurcation diagrams or basin boundaries, a lower level compiler-based programming environment such as Pascal or C is more suitable. There are also several software packages available for more advanced numerical analysis, such as computing stable and unstable manifolds and computing bifurcation curves (so-called continuation software). Examples are *Dynamics 2* (Nusse and Yorke), *DSTool* (Guckenheimer, a.o.), *Auto2000* (Doedel et al. 2001) and *Content* (Kuznetsov). Some of the numerical procedures used in these software packages are based on advanced mathematical results concerning numerical detection of manifolds (e.g. Nusse and Yorke 1998) and bifurcation curves (e.g. Kuznetsov 1995).

The *E&F Chaos* program is a stand-alone Windows software package for the *initial numerical exploration* of dynamical systems formulated by students or researchers. In terms of functionality and implementation, *E&F Chaos* is perhaps most closely related to the *iDMC* package developed by the universities of Udine and Ca'Foscari of Venice as part of a teaching unit, which also includes a textbook containing many examples and exercises (Medio and Lines 2001). Both *iDMC* and *E&F Chaos* simulate dynamical systems using the LUA language, in which the user specifies the model. The graphical user interface of *iDMC* is implemented in Java, which has the advantage of being platform independent (a Windows as well as a Linux version are available), but the user has to install the Java Runtime Environment as well.[2] *E&F Chaos* and *iDMC* share many functional features, but differ in the way the user specifies the details and the way in which the results are presented.

E&F Chaos provides the user with default choices for the parameters to be specified for each of the methods, endorsing the philosophy to provide a quick initial analysis of the dynamics of a specified model. To include a new model for simulations, the

---

[2] E&F Chaos can be run under the Wine translation layer on Linux systems.

user only has to specify the dynamics in the flexible LUA language, as a text file. If desired, auxilary variables for which output (e.g. a time series) is required can be defined and included in the specification. Prior analytic calculations, such as computation of a Jacobian matrix, are not necessary. For example, in order to compute the largest Lyapunov exponent *E&F Chaos* evaluates the Jacobian numerically from the specified map, rather than from an analytic expression of the Jacobian matrix; see Subsect. 4.5 for more details. To allow further processing of the data generated, *E&F Chaos* allows the data represented in any graph to be written to file in plain text format. *E&F Chaos* provides graphical output in a range of formats, including bitmap, Windows Metafile and Enhanced Metafile. It is possible to generate publication quality output of graphs as well, in which case *gnuplot* will be called to generate an Encapsulated Postscript (.EPS) file. To keep the style and details such as axes labels of these high-quality graphs flexible, there is an option to save the *gnuplot* file as well as a data file in plain text format, which can be edited and processed by *gnuplot* later if desired. All figures in the current paper have been directly generated by *E&F Chaos*.[3]

3 One-Dimensional Example

To illustrate simulation of a 1-D system, we consider a simple economic example, the cobweb model with adaptive expectations. The classical cobweb model is a partial equilibrium model describing price fluctuations of a non-storable consumption good. There is a one period lag in production, so producers have to form price expectations one period ahead. Demand, supply and market clearing are given by

$$D(p_t) = a - dp_t, \quad a \in R, \quad d \geq 0 \tag{1}$$

$$S_\lambda(p_t^e) = \arctan(\lambda p_t^e), \quad \lambda > 0, \tag{2}$$

$$D(p_t) = S_\lambda(p_t^e). \tag{3}$$

Demand D is a linearly decreasing function in the market price p_t, with slope $-b$. Following Chiarella (1988) and Hommes (1994), we consider a *nonlinear*, increasing, S-shaped supply curve S_λ, where the parameter λ tunes the nonlinearity of the supply curve.[4]

To close the model we have to specify how producers form price expectations. The simplest case, studied in the thirties e.g. by Ezekiel (1938), assumes that producers

---

[3] Encapsulated Postscript (EPS) files can be viewed and processed by many software packages, such as Ghostview (http://pages.cs.wisc.edu/ghost/), GNU gv (http://www.gnu.org/software/gv/), and can be imported by document typesetting software such as LaTeX.

[4] We use the same specification as in Hommes (1994), with the inflection point of the supply curve chosen as the origin. Negative values thus mean negative deviations from this inflection point.

have naive expectations, that is, their prediction equals the last observed price $p_t^e = p_{t-1}$. In his seminal paper Muth (1961) used the framework of the cobweb model to introduce rational expectations, with the price forecast $p^e = p^*$, where p^* is the price corresponding to the intersection point of demand and supply. Here we consider the case of adaptive expectations, proposed by Nerlove (1958). In contrast to Nerlove (1958), we have a non-linear, instead of a linear, supply curve. *Adaptive expectations* are given by

$$p_t^e = (1 - w)p_{t-1}^e + wp_{t-1}, \quad 0 \le w \le 1, \tag{4}$$

where w is the expectations weight factor. The expected price is a weighted average of yesterday's expected and realized prices, or equivalently, the expected price is adapted by a factor w in the direction of the most recent realization. A simple computation, using (1–3) and (4), shows that the dynamics of expected prices becomes

$$p_t^e = (1 - w)p_{t-1}^e + w\frac{a - \arctan(\lambda p_{t-1}^e)}{b} = f_{w,a,b,\lambda}(p_{t-1}^e). \tag{5}$$

Dynamics of (expected) prices in the cobweb model with adaptive expectations is thus given by a 1-D system $x_t = f_{w,a,b,\lambda}(x_{t-1})$ with four model parameters. We next discuss how this model can be simulated in *E&F Chaos*.

3.1 Getting Started with E&F Chaos

After installing the software package *E&F Chaos* two directories are created: 'gnuplot' (containing all *gnuplot* files) and 'Models' (containing all model files). The models are collected in three subdirectories, 'Continuous' and 'Discrete' (containing some benchmark examples of nonlinear differential and difference equations, respectively), and 'My Models', containing newly created models by the user. When running the software, the *E&F Chaos* window appears, with an easy-to-use Windows menu structure. To simulate the model of this section, the user can use the File/Load Model menu, open the subdirectory 'My Models' and select the file 'Cobweb Adaptive Expectations' to run simulations. All menu/submenu items are also available as panel buttons at the top of the *E&F Chaos* window.

3.2 Time Series

The most important menu in the *E&F Chaos* software is the Plot menu, where all graphical plot submenus are collected, which will be illustrated throughout this paper. Chiarella (1988) and Hommes (1991, 1994) have shown that the cobweb model with a linear demand curve and a *nonlinear*, but monotonic, supply curve can generate chaotic fluctuations. Using the Time Series submenu, an example of a chaotic price series is illustrated in Fig. 1. Each graphical window, such as the 'Time Series' window, has a submenu 'Edit', which enables the user to change parameter values, the initial state, the plot variable (only the expected price 'x' in this example), the number of points
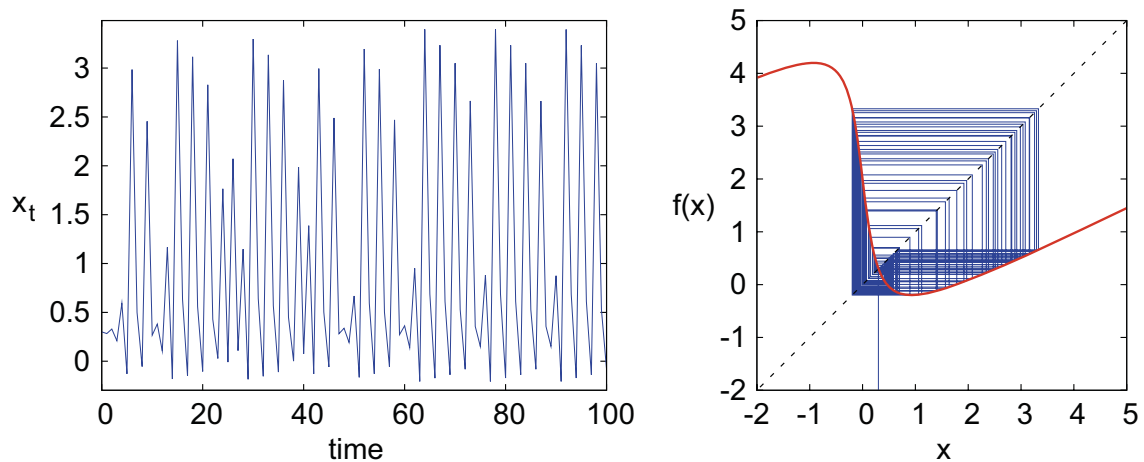
Fig. 1 Chaotic time series (left panel) for expectations weight factor $w = 0.5$, with other parameter values $\lambda = 4.4$, $a = 1$ and $b = 0.25$ and initial state $x_0 = 0.3$. A graphical analysis (right panel) shows that the map $f_{w,a,b,\lambda}$ is non-monotonic with two critical points, with a maximum and a minimum, and that initial state $x_0 = 0.3$ does not converge to a low periodic orbit

plotted, the number of points not plotted (so that a transient phase can be skipped) and the time step (so that e.g. each 10th period can be plotted). The Time Series menu has the additional option 'Connect Points' (if on, then points are connected by a line) and 'Plot Squares' (if on, then each point is represented by a square). Scaling can be done either automatically (default) or manually.

3.3 Graphical Analysis

One-dimensional (1-D) systems $x_t = f(x_{t-1})$ are relatively easy to analyze, since the graph of the corresponding 1-D generating map f essentially determines the dynamical behavior. The Graphical Analysis submenu (only available for 1-D systems) gives the possibility for detailed investigations of the graph of the map f and higher order iterates f^k, $k \geq 2$ (i.e. applying the map f k times). Figure 1 (right panel) shows that the graph of the map $f_{w,a,b,\lambda}$ in (5) is a non-monotonic map, with two critical points where the graph has a (local) maximum and a (local) minimum, respectively. In the Settings submenu, the user can specify the range of the interval (i.e. minimum and maximum), the number of iterates and an initial state. By clicking 'Next Iteration(s)' the next steps in the graphical analysis (i.e. a horizontal line segment connecting the last point to the diagonal $y = x$, and a vertical line segment connecting this point to the graph of the map) appear. By repeatedly clicking 'Next Iteration(s)' a 'cobweb diagram' as in Fig. 1 (right panel) appears, illustrating the dynamical behavior. Since the graphical analysis in this example does not converge, it suggests that the dynamical behavior is chaotic. After clicking 'Clear Form', the graphical analysis can be repeated. Using the Edit submenu, the user can specify the number of time steps as an integer k (so that one click to 'Next Iteration(s)' yields k steps in the graphical analysis). In Subsect. 3.6 we will return to the 'Graphical Analysis' to investigate bifurcations in this model.

3.4 Bifurcation Diagram

A powerful tool for investigating how the dynamical behavior of a nonlinear model depends on a single parameter is a *bifurcation diagram*. A bifurcation is a qualitative change in the dynamics as a model parameter changes. For instance, a fixed point becomes unstable if one or more eigenvalues of the linearized dynamics around the fixed point cross the unit circle. The three possible scenarios in which that may happen are (see e.g. Kuznetsov 1995 for a mathematical treatment of bifurcation theory):

- an eigenvalue $\lambda = +1$: typically this corresponds to a *saddle node* or *tangent* bifurcation from 0 to 2 steady states. Other possibilities are a *pitchfork* bifurcation (from 1 to 3 steady states) or a *transcritical* bifurcation (exchange of stability of 2 steady states);
- an eigenvalue $\lambda = -1$: *period doubling* or *flip* bifurcation, with creation of a 2-cycle;
- a complex pair of eigenvalues with $\|\lambda\| = 1$: Hopf or Neimark–Sacker bifurcation. This possibility only arises for two or higher dimensional systems.

A bifurcation diagram shows the long run dynamical behavior as a function of a model parameter. After selecting the Bifurcation Diagram menu, the user can specify (or use the default values) the parameter, its minimum and maximum value, the plot variable, the number of points to be plotted and the transient time (i.e. the number of points skipped before plotting is started). To provide different ways of dealing with the possibility of multiple attractors, there are several options for the initial conditions for different parameter values as the bifurcation diagram is computed. There are three options: (i; default option) initialize on the same, fixed initial condition for each new parameter value, (ii) initialize on the last point plotted for the previous, slightly smaller parameter value, (iii) compute the diagram from 'right to left', i.e. from high to low values, with initial states as in (i) or (ii). Options (ii) and (iii) are intended to keep the orbit on the 'same' attractor as the parameter slowly changes, thus following the attractor as long as it exists.

Figure 2 shows a bifurcation diagram of the cobweb model with adaptive expectations with respect to the expectations weight factor w, illustrating the long run dynamics (100 iterations) after omitting a transient phase of 100 iterations. For small values of w, $0 \leq w \leq 0.27$, prices converge to a stable steady state, while for high values of w, $0.77 < w \leq 1$ (close to naive expectations) prices converge to a stable 2-cycle with large amplitude. For intermediate w-values however, say for $0.4 < w < 0.7$, chaotic price oscillations of moderate amplitude arise. In particular, the chaotic price fluctuations for $w = 0.5$ have been illustrated already in Fig. 1.

3.5 Parameter Basins

An even more powerful tool in the numerical analysis of nonlinear dynamics are the *Parameter Basin* plots, sometimes also called 2-D bifurcation diagrams. A parameter basin plot assigns different colors in a 2-D parameter space to stable cycles of different periods. The parameter basins are computed as follows. For the selected parameters

Fig. 2 Bifurcation diagram w.r.t. the expectations weight factor w, $0.2 \leq w \leq 0.9$, with the other parameters fixed at $a = 1$, $b = 0.25$ and $\lambda = 4.4$. The diagram shows the long run behavior (100 iterations) after a transient of 100 iterations
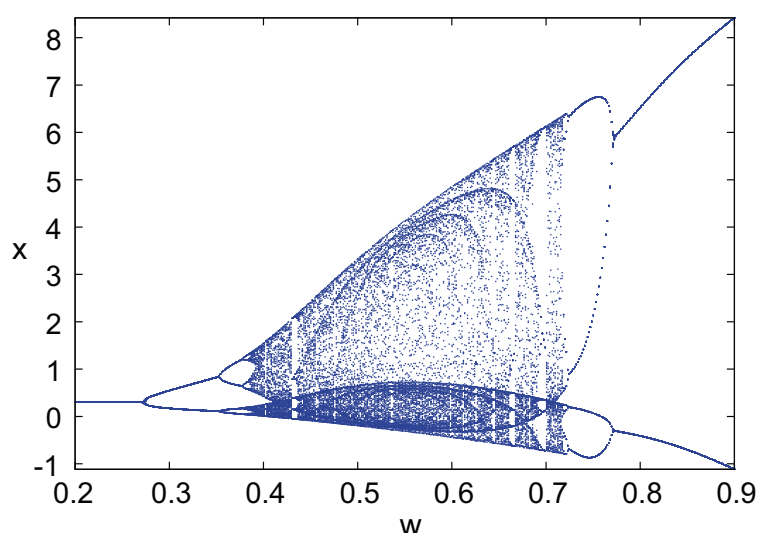
from a 2-D grid a trajectory for the selected initial state is chosen. After a transient period K, a starting point on the trajectory is fixed. By iterating the map from this starting point convergence to a cycle of a specified period is verified. Convergence to a period P cycle is assumed to have occurred when the Euclidean norm between the starting point and the point reached after P iterations is less than some specified convergence distance, epsilon.

Using the Parameter Basins menu in *E&F Chaos*, the user can specify the number of periodic cycles to look for, the distance of convergence (epsilon), the two parameters, the number of transient iterations, the maximum distance (this is the distance after which the point is classified as "divergence", in order to prevent overflows), the periods of the cycles to be detected, the minimum and maximum of the two parameter intervals and the resolution for both intervals. Finally, the user can specify the colors corresponding to each of the different stable cycles, the color for the non-convergence parameters (i.e. parameters for which points did not converge within distance epsilon) and the color for divergence parameters (i.e. parameters for which the trajectory diverges to infinity). For an initial run of the parameter basins, low values of the resolution (e.g. 200) and the transient time (e.g. 100) are recommended, because computation time rapidly increases with higher resolution. The default resolution is set to the maximum screen resolution. After clicking the 'Run' button, the computations start and a progress bar showing the fraction of computed points appears. The user can interrupt the computations by clicking the 'Stop' button.

Figure 3 presents a parameter basin with respect to the parameters (λ, w), and assigns different colors to stable steady states and stable cycles of periods 2, 4, 8 (the first four cycles in a period doubling bifurcation route to chaos) and periods 3, 5 and 7 (low order stable cycles of odd period). It is well known that, for 1-D continuous maps, a cycle with odd period implies chaotic dynamical behavior.[5]

---

[5] An odd cycle implies so-called *topological chaos*. In such a system however, chaos may still be a transient phenomena as a typical time series may still converge to a stable cycle. As a special case, the famous "Period 3 implies chaos" result of Li and Yorke (1975) holds; see e.g. Devaney (1989) for a detailed mathematical treatment.
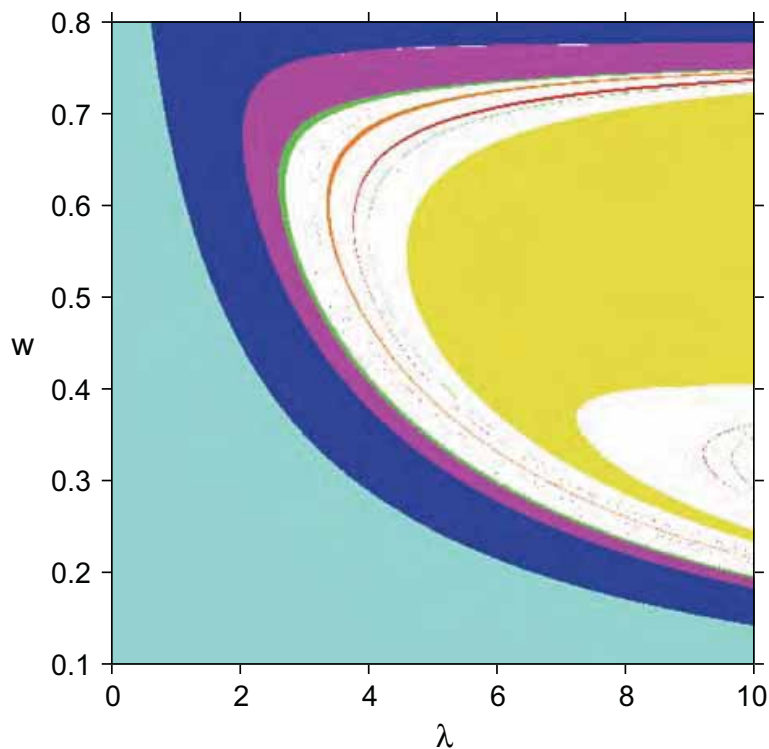
Fig. 3 Parameter Basins of stable steady state (light blue) and stable cycles of periods 2 (dark blue), 4 (purple), 8 (green), 3 (yellow), 5 (orange) and 7 (red), for $0 \leq \lambda \leq 10$ and $0.1 \leq w \leq 0.8$ with other parameters fixed at $a = 1$ and $b = 0.25$ and initial state $x_0 = 0.3$. Transient iterations 500, resolution 450×450 and convergence distance epsilon 0.001. Computation time 165 s (Pentium 3 GHz)

Notice that the 1-D bifurcation diagram with respect to the parameter w, as illustrated in Fig. 2, corresponds to a vertical line in the parameter basin plot, with fixed $\lambda = 4.4$, in Fig. 3. The reader should compare the bifurcation scenario's in both figures.

3.6 Explaining Bifurcations Through Graphical Analysis

The boundaries between different colors in the parameter basins in Fig. 3 represent bifurcation curves in the 2-D (λ, w)-parameter space. For 1-D systems, bifurcations of cycles can be easily investigated by the Graphical Analysis menu. For example, Fig. 3 suggest that fixing $\lambda = 6$ and increasing w leads to a *period doubling bifurcation* when the dynamics changes from a stable steady state (the light blue region) to a stable 2-cycle (the dark blue region). The graphical analysis in Fig. 4 confirms this conjecture. At the bifurcation value $w \approx 0.21$ (top left), the graph of the second iterate f^2 is tangent to the diagonal at the steady state, and the steady state is their only intersection point. Beyond the bifurcation value, as illustrated for $w = 0.25$ (top right), the graph of f^2 has three intersection points with the diagonal, namely the steady state and two additional points forming a (stable) 2-cycle. The parameter basins plot in Fig. 3 suggests that more bifurcations occur when w is further increased. The reader may check e.g. that a second period doubling bifurcation occurs for $w \approx 0.28$, by looking at the graphs of the fourth iterate
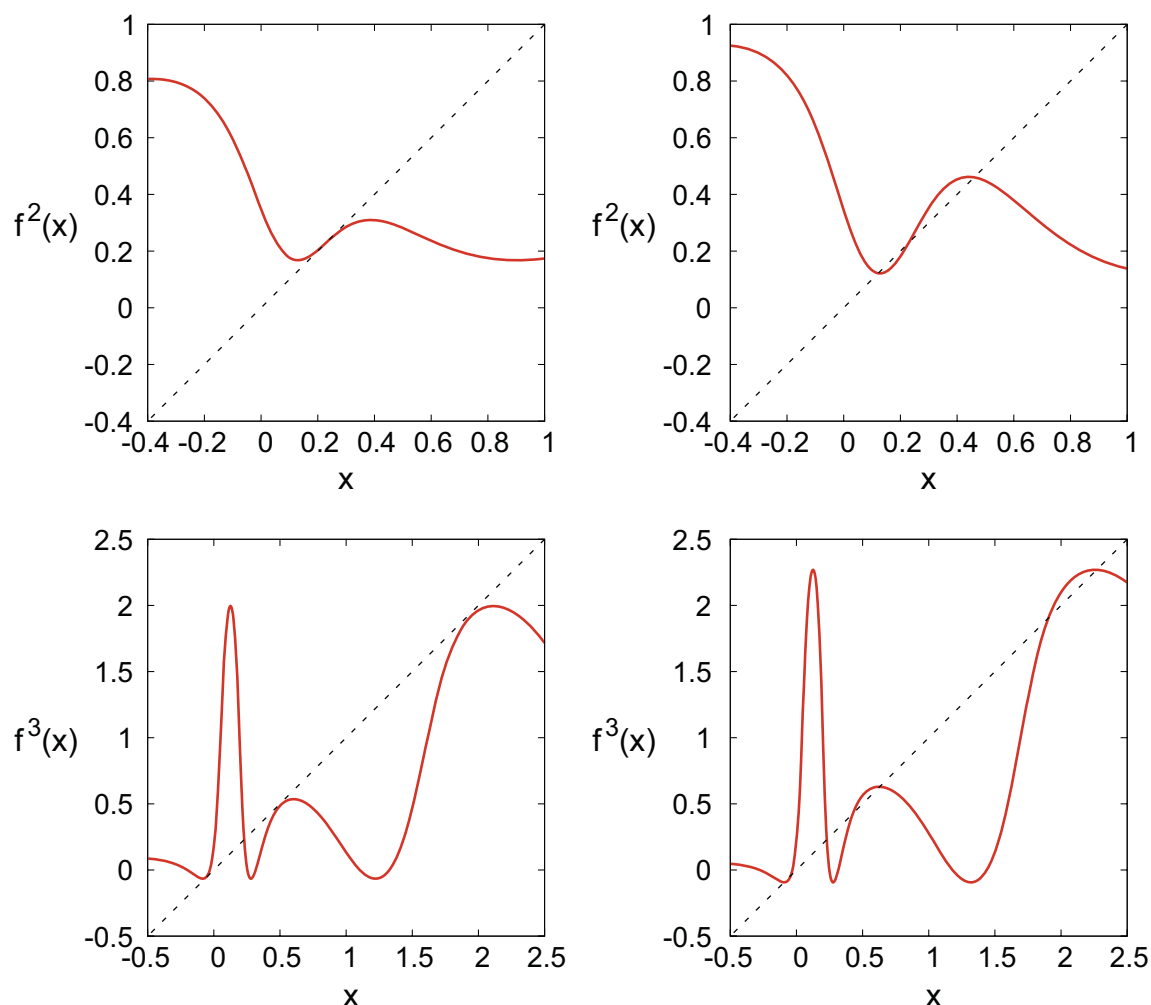
Fig. 4 Graphical analysis of bifurcations occurring when w increases, with other parameters fixed at $\lambda = 6$, $a = 1$ and $b = 0.25$. A period doubling bifurcation from a stable steady state to a stable 2-cycle occurs for $w \approx 0.21$, when the graph of f^2 is tangent to the diagonal (top left). For $w = 0.25$ the graph of f^2 has three intersection points with the diagonal, one steady states and two points of a (stable) 2-cycle (top right). A tangent bifurcation of a pair of 3-cycles occurs for $w \approx 0.37$, where the graph of f^3 is tangent to the diagonal (bottom left). For $w = 0.385$ the graph of f^3 has seven intersection points with the diagonal, the steady state and six additional points forming one stable and one unstable 3-cycle (bottom right)

f^4. A different type of bifurcation occurs when entering the yellow region, corresponding to a stable 3-cycle. The 3-cycle is created by a *tangent bifurcation* for $w \approx 0.37$, as illustrated in Fig. 4 (bottom left), where the graph of f^3 becomes tangent to the diagonal simultaneously at the three points of a newly created 3-cycle. More precisely, a pair of two 3-cycles is created in the tangent bifurcation, one stable and one unstable, as illustrated for $w = 0.385$ in Fig. 4 (bottom right). The reader may also check that increasing w further, another tangent bifurcation in which both 3-cycles disappear occurs for $w \approx 0,67$. In fact, changing w (or some other parameter) many bifurcations, period doubling and/or tangent bifurcations, occur for higher order iterates f^k of the map f. The user can easily investigate these with *E&F Chaos* by setting the value of 'Iterated Map' in the Settings submenu to k.

🐎 Springer

4 A Higher Dimensional Example

In this section we consider a higher dimensional example of a nonlinear system, to illustrate more features of the *E&F Chaos* software. The model is an extension of the agent-based cobweb model with heterogeneous expectations (rational versus naive expectations) introduced by Brock and Hommes (1997), modified to allow for *asynchronous* updating of strategies. This extension allows us to illustrate how easy it is to include a new or extended model in *E&F Chaos*. As in the previous section, demand is linear and given by

$$D(p_t) = a - bp_t. \tag{6}$$

The supply is also assumed to be linear, derived from expected profit maximization from a quadratic cost function $c(q) = q^2/(2s)$, and therefore given by

$$S(p_{ht}^e) = sp_{ht}^e, \tag{7}$$

where p_{ht}^e is the expected market price by agents of type h. Agents can choose between two different types of expectation rules, a costly sophisticated rule and a cheap free rule. The type 1 rule is rational expectations, which can be obtained at information gathering costs $C > 0$, while the type 2 rule is naive expectations, which is available at no cost. The forecasting rules thus are

$$p_{1t}^e = p_t, \qquad p_{2t}^e = p_{t-1}. \tag{8}$$

With rational and naive producers, the market clearing condition is given by

$$a - bp_t = n_{1t}sp_t + n_{2t}sp_{t-1}, \tag{9}$$

where n_{1t} and n_{2t} represent the fractions of type 1 and type 2 agents. The unique market clearing price p_t is then given by

$$p_t = \frac{a - n_{2t}sp_{t-1}}{b + sn_{1t}}. \tag{10}$$

The fractions are determined by the past performance of the two forecasting strategies. The payoff π_{ht} of strategy h at time t is given by the realized profit of that strategy at time t minus the costs incurred. Using producers' quadratic cost function $c(q) = q^2/(2s)$ and taking the constant information gathering costs $C > 0$ for rational expectations into account, one finds

$$\pi_{1t} = \frac{b}{2}p_t^2 - C, \qquad \pi_{2t} = \frac{b}{2}p_{t-1}(2p_t - p_{t-1}). \tag{11}$$

The fitness measure U_{ht} of strategy h is a geometrically downweighted average of past profits, i.e.

$$U_{ht} = wU_{h,t-1} + (1-w)\pi_{ht}, \tag{12}$$

where $w \in [0, 1)$ is a memory parameter. The extreme case $w = 0$ represents agents basing their evaluation only on the profits observed in the previous period. Brock and Hommes (1997) considered this model with *synchronous* updating of strategies, that is, in each period *all* agents update their strategies. Here we consider the more general case of *asynchronous* updating. Per time unit only a fraction $1 - \delta$ of agents, distributed randomly among agents of both types and independently across time, is assumed to reconsider their strategy on the basis of the most recent information available. The remaining fraction δ continues to use their previous strategy. The corresponding dynamics of the fractions is given by a modified version of the discrete choice, logit probabilities:

$$n_{ht} = (1-\delta)e^{\beta U_{h,t-1}}/Z_{t-1} + \delta n_{h,t-1}, \tag{13}$$

where $Z_{t-1} = \sum_h e^{\beta U_{h,t-1}}$ is a normalization factor so that the fractions add up to 1. For $\delta = 0$, we are back in the case of synchronous updating. In evolutionary game theory there has been a discussion whether asynchronous updating may lead to more stability (cf. Nowak and May 1992; Huberman and Glance 1993; Nowak et al. 1992). Financial market models with asynchronous updating have been considered by Diks and van der Weide (2005) and Hommes et al. (2005).

The dynamics of the cobweb model with rational versus naive agents and asynchronous updating of strategies is thus given by Eqs. 10–13, defining a nonlinear four-dimensional (with state variables p_{t-1}, $n_{1,t-1}$, $U_{1,t-1}$ and $U_{2,t-1}$) discrete time dynamical system. A typical research question for such a generalization of an existing model is how the extra parameters affect the dynamics. Our aim is to illustrate how the *E&F Chaos* program can help to obtain a quick overview of the (long-run) behavior of the dynamics in different regions of the parameter space.

4.1 Building a New Model in E&F Chaos

Including a new model or modifying an existing model in the *E&F Chaos* software is easy. All the user has to do is to specify the model in LUA code (or modify an existing model) in a plain text file, either with the built-in editor or with her own favorite editor.[6] To view an example, the user can use the File/Load Model submenu, open the subdirectory 'My Models' and select the file 'BH-cobweb-async', the cobweb model with asynchronous updating as presented in the previous subsection. Using the File/Edit Model submenu, a window containing the model equations appears:

---

[6] For more information about the LUA language, see the LUA website www.lua.org.

```
p=2, n1=0.5, n2=0.5, u1=0, u2=0
beta=5, a=4, b=0.5, s=1.35, C=1, w=0,
delta=0
pchange
d
-- BH-cobweb-async

-- lagged variables
plag = p
n1lag = n1
u1lag = u1
u2lag = u2


-- price
p = (a-n2*s*plag)/(b+n1*s)

-- fitness
u1 = w*u1lag + (1-w)*((b/2)*p*p-C)
u2 = w*u2lag + (1-w)*(b/2)*plag*(2*p-plag)

-- fractions
q1 = math.exp(beta*u1)
q2 = math.exp(beta*u2)
z = q1+q2
n1 = delta*n1lag + (1-delta)*q1/z
n2 = 1-n1

-- price change
pchange=p-plag
```

Strictly speaking the first four lines are not LUA code. They comprise a four line header to inform the *E&F Chaos* program about the various model parameters, state variable and the type of model. The first line provides the state variables (p, n_1, n_2, u_1 and u_2) and their default initial states. The second line specifies the model parameters (the intensity of choice β, the constant a and slope b of the demand curve, the slope s of the supply curve, the information gathering costs C, the memory w in the fitness measure and the parameter δ measuring asynchronous updating) as well as their default values. Any additional variables (pchange, the price change) of which output is desired are provided on the third line, while the fourth and final header line contains a letter to specify whether we are dealing with a continuous time ('c') or discrete time ('d') system. As an alternative to providing either 'c' or 'd' on the fourth line of the header, at the bottom of the Edit Model window, the user can specify the 'Type' (i.e. either 'c' for a continuous time model or 'd' for discrete time). The remaining code is used to describe the dynamical system by specifying the new values of the state
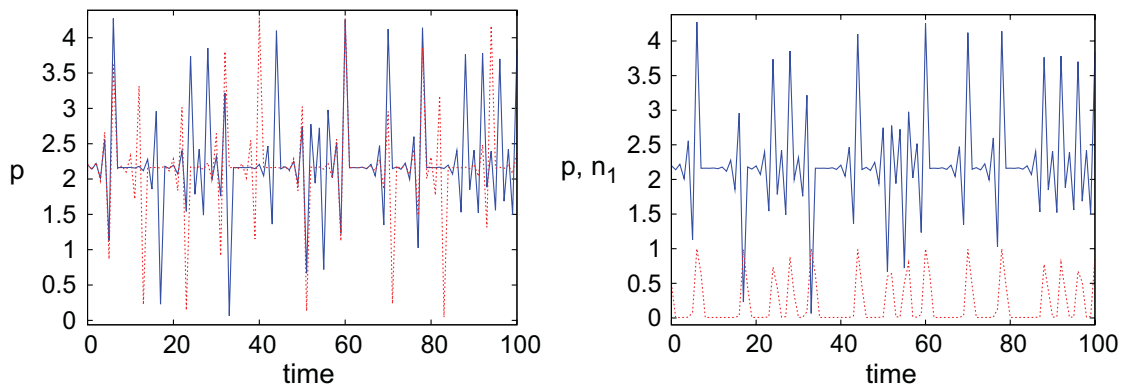
Fig. 5 Multiple times series. Time series for slightly different initial conditions (left panel) $x_0 = 2.2$ (solid line) and $x_0 = 2.21$ (dotted line), illustrating the sensitive dependence on initial conditions. Simultaneous time series of prices and fraction of rational agents (right panel). The fraction of rational agents increases when price deviations from steady state become large. Parameters are: $\beta = 5$, $a = 4$, $b = 0.5$, $s = 1.35$, $C = 1$, $w = 0$ and $\delta = 0$

variables in terms of the lagged values of the state variables and the parameter values. Since this code is written in the LUA language, one can use standard prespecified functions in LUA, such as the exponential function (`math.exp()`) used here. The user can include some comments, i.e. text following $--$, to make the code more easy to read. A convenient feature for applied users, such as economists, is that there is no need to reduce the model equations to its simplest mathematical representation of a (4-D) difference equation $X_{t+1} = F(X_t)$. The user is free to use other representations, e.g. following the same economic principles as in the presentation of the model in the previous section, i.e. using the pricing equation (10), the fitness measure (11–12) and the fractions (13). Of course, a shorter, more efficient representation may lead to a reduction of computation time, but for relatively simple models this is not critical.

4.2 Multiple Time Series

A convenient feature of the *E&F Chaos* software is that it can simultaneously plot different time series in a single graphics window, as illustrated in Fig. 5. One can use this feature, for example, to investigate and illustrate whether the model exhibits *sensitive dependence on initial conditions*, a key feature of a chaotic system. Figure 5 (left panel) shows simultaneous plots of price sequences with slightly different initial conditions. Since the two time series separate already after about 10 periods, the system exhibits sensitive dependence. Simultaneous time series are also useful in understanding the (economic) mechanism causing cycles and fluctuations. As an example, Fig. 5 (right panel) shows simultaneous plots of the price series and the fraction of rational agents. The fraction of rational agents quickly increases, each time price deviations from the steady state become too large. A large fraction of rational agents pushes the price back close to the steady state, causing agents to abandon the costly rational forecast and switch back to the cheap naive rule.
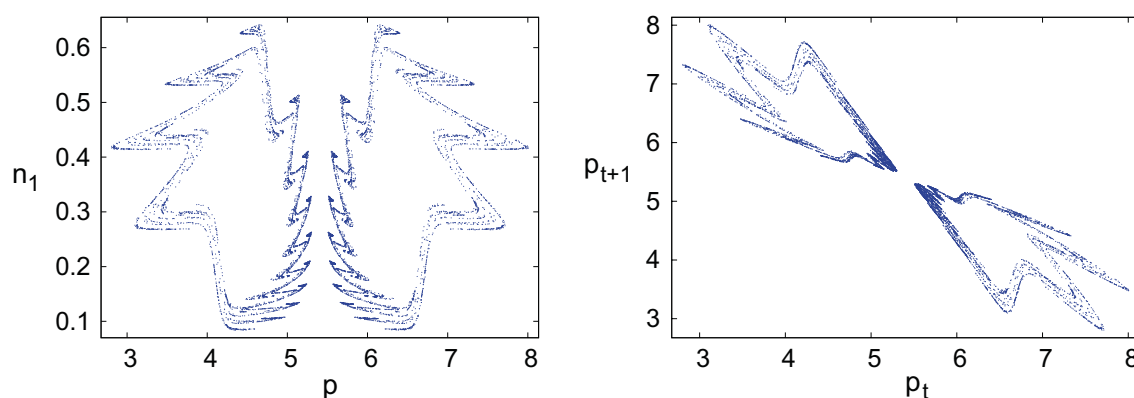
Fig. 6 Phase plot (left panel) with a strange attractor in the (p, n_1) plane. The same attractor shown in a delay plot (p_t, p_{t+1}) (right panel). Parameters are: $\beta = 15$, $a = 10$, $b = 0.5$, $s = 1.35$, $C = 1$, $w = 0$ and $\delta = 0.8$

4.3 Phase Plots and Delay Plots

For higher dimensional systems, Phase Plots and Delay Plots provide helpful tools to investigate the (long run) dynamical behavior and plot attractors of the system. Intuitively, an attractor of a dynamical system is the subset of state space points to which the dynamics are confined in the long run. The attractor thus contains information about the long run dynamical behavior of the system. Examples are a stable steady state (single point attractor), a stable periodic orbit (finite number of points), a quasi-periodic attractor ("invariant circle") and a strange (chaotic) attractor (fractal structure). Examples of a phase plot and a delay plot of a strange attractor are shown in Fig. 6. By holding the mouse button and drawing a rectangle, the user can zoom into the detailed fractal structure of the attractor.

4.4 Iterating Objects: The Unstable Manifold of the Steady State

The attractors in Fig. 6 correspond to the case of asynchronous updating ($\delta = 0.8$) and no memory in the fitness measure ($w = 0$). In this case, mathematically the system can be reduced to a 2-D system. In the case of synchronous updating of strategies ($\delta = 0$), Brock and Hommes (1997) have shown that, for large values of the intensity of choice β the system is close to a homoclinic tangency between the stable and unstable manifolds of the steady state. The strange attractor in Fig. 6 suggests that a near homoclinic tangency may also explain the complicated dynamics in the case of asynchronous updating of strategies. A mathematical analysis is beyond the scope of this paper (for a mathematical treatment of homoclinic bifurcations see e.g. Palis and Takens 1993), but we can use the Iterating Objects submenu to approximate the unstable manifold of the steady state. The user can iterate circles as well as polygons over the attractor. Iterating small circles can e.g. illustrate the sensitive dependence and how quickly nearby points spread over a strange attractor. The option of Iterating a polygon and setting the number of vertices to two enables the user to iterate vectors. This option can then be used to iterate a small unstable eigenvector originating at the steady state thus approximating the unstable manifold of the steady state, as illustrated
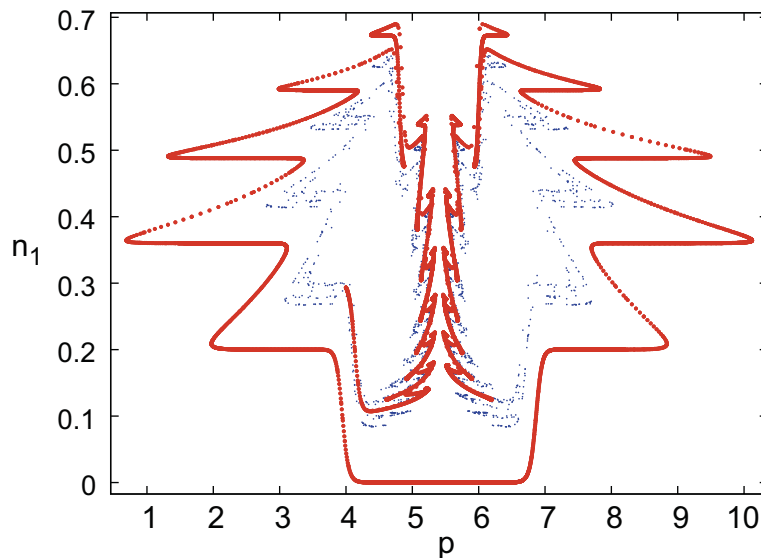
Fig. 7 Approximation of the unstable manifold of the steady state; $1,000$ point of the strange attractor are also shown. The approximation of the unstable manifold has been obtained by 19 iterations of $1,000$ initial states on a small vector with endpoints $(p^*, 0)$ and $(p^* + 0.001, 0)$. Parameters are: $\beta = 15$, $a = 10$, $b = 0.5$, $s = 1.35$, $C = 1$, $w = 0$, $\delta = 0.8$ and $p^* = a/(b + s) \approx 5.4054$

in Fig. 7. The user can e.g. specify the endpoints and the number of initial states (the resolution) on the initial vector. The reader may easily check that the stable manifold of the steady state is a vertical line $p = p^*$. Figure 7 shows that for large values of the intensity of choice, the system is indeed close to a homoclinic tangency since the unstable manifold gets close to the stable manifold. In the next section we investigate numerically whether, in the case of asynchronous updating, chaos arises.

4.5 Largest Lyapunov Exponent

A powerful numerical tool to investigate whether the dynamical behavior is chaotic is a plot of the largest Lyapunov exponent, as a function of one of the model parameters. The largest Lyapunov exponent is the average growth rate of an infinitesimal state perturbation along a typical trajectory (orbit). *E&F Chaos* calculates the largest Lyapunov exponent along the lines of the algorithm introduced by Wolf et al. (1985). Let $\| \cdot \|$ denote the Euclidean norm. The distance in state space between two initially close state vectors $\boldsymbol{x}_0$ and $\boldsymbol{x}_0 + \boldsymbol{\delta}$, for small $\|\boldsymbol{\delta}\|$ evolves as

$$F^n(\boldsymbol{x}_0 + \boldsymbol{\delta}) - F^n(\boldsymbol{x}_0) \approx DF^n(\boldsymbol{x}_0)\boldsymbol{\delta} = DF(\boldsymbol{x}_n) \cdot DF(\boldsymbol{x}_{n-1}) \dots DF(\boldsymbol{x}_1)\boldsymbol{\delta},$$

where $DF(\boldsymbol{x})$ denotes the Jacobi matrix of F evaluated at state $\boldsymbol{x}$. For typical initial states $\boldsymbol{x}_0$ the length of the perturbation grows exponentially as

$$\|DF^n(\boldsymbol{x}_0)\boldsymbol{\delta}\| \sim e^{\lambda n}\|\boldsymbol{\delta}\|,$$

motivating a definition of largest Lyapunov exponent as

$$\lambda = \lim_{n \to \infty} \frac{1}{n} \log \frac{\|DF^n(x_0)\delta\|}{\|\delta\|}.$$

4.5.1 Algorithm for Lyapunov exponent

Generally speaking, λ can be estimated numerically either by specifying the Jacobi matrix analytically, evaluating it at any point along a generated orbit, or by using a finite differencing method along the orbit. Wolf et al. (1985) opt for the first method in case the dynamics are known, while they use the second approach in case the dynamics is reconstructed from an observed time series. We use a finite difference approach also in the case of known dynamics, thus avoiding an error prone specification of the Jacobi matrix by the user. The price payed is a slight loss of accuracy due to the finite difference approximation and numerical noise, but for most practical purposes the differences are sufficiently small, especially when the number of iterations is large (e.g. about 10,000).

The main idea behind the finite differencing algorithm for the largest Lapunov exponent is to follow the evolution of a small initial disturbance of the state around a reference trajectory (assumed to be typical orbit generated by the dynamics). To prevent the distance between the disturbed and the reference trajectory from becoming too large, this distance is regularly (in our implementation every time step) rescaled to a small number ε while keeping the direction of the disturbance unaffected. In pseudocode:

Set $r = 0$
Take an initial vector x_0 and a perturbed vector $x + \delta_0$ with $\|\delta_0\| = \varepsilon$.
For $t = 1, \ldots, n$
{
    iterate both states: $x_t = F(x_{t-1})$ and $x_t + \delta'_{t-1} := F(x_{t-1} + \delta_{t-1})$
    Sum up the growth rate along the orbit: $r = r + \log \frac{\|\delta'_{t-1}\|}{\|\varepsilon\|}$
    Rescale the perturbation δ'_{t-1} to length ε through $x_t + \delta_t := x_t + \frac{\|\varepsilon\|}{\|\delta'_{t-1}\|}\delta'_{t-1}$.
}
Estimate λ as te average growth rate r/n.

The perpetual rescaling of the perturbation is used to keep the distance between the reference and the perturbed trajectory small. In this way the evolution of a tangent vector along the trajectory is approximated, allowing the difference between the two trajectories to evolve towards the most unstable direction of the dynamics, which is the direction associated with the largest Lyapunov exponent. The initial disturbance is chosen to be a positive change of size ε of the first element of the state variable.

Figure 8 shows a bifurcation diagram and a Lyapunov exponent plot with respect to the intensity of choice parameter β, in the case of asynchronous updating of strategies ($\delta = 0.8$). The bifurcation diagram (top panel) shows a primary period doubling bifurcation (from a stable steady state to a stable 2-cycle), a secondary Hopf bifurcation of the 2-cycle (the reader may check this by a phase plot of the attractor for e.g. $\beta = 3$,
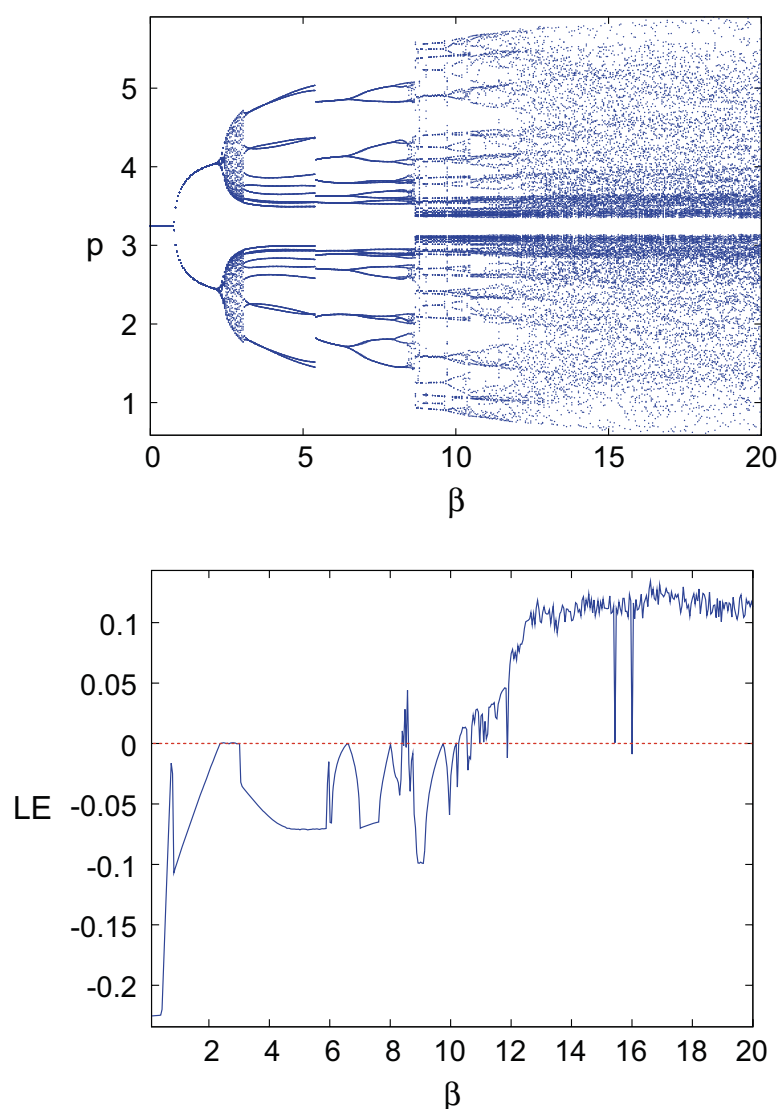
Fig. 8 Bifurcation diagram (top panel) and largest Lyapunov exponent (LE) plot (bottom panel) in the case of asynchrounous updating of strategies. The bifurcation diagram shows the long run dynamical behavior, 100 iterations after a transient of 100 iterations, as a function of the parameter β. The LE plot is based on 5000 iterations after a transient of 300 iterations and takes 35 s (Pentium 3 GHz). Other parameters are: $a = 6, b = 0.5, s = 1.35, C = 1, w = 0$ and $\delta = 0.8$

showing the invariant circles created in the Hopf bifurcation), followed by a bifurcation route to more complicated dynamical behavior. The Lyapunov exponent plot (bottom panel) becomes positive for large values of β, providing numerical evidence that the dynamics becomes chaotic for large values of the intensity of choice. These simulations thus show that the rational route to randomness, that is, the bifucation route to chaos, found by Brock and Hommes (1997) in the cobweb model with synchronous updating of strategies also arises in the case of asynchronous updating of strategies.

Using the Largest Lyapunov Exponent submenu, the user can specify the parameter, the range of the parameter, the size of the perturbation vector (default at 0.001), the number of iterations and the transient time for computation of the Lyapunov exponents. The output is given as a graph of the estimated largest Lyapunov exponent as a function of this parameter range. If required the resolution in the parameter space can be adjusted by the user.
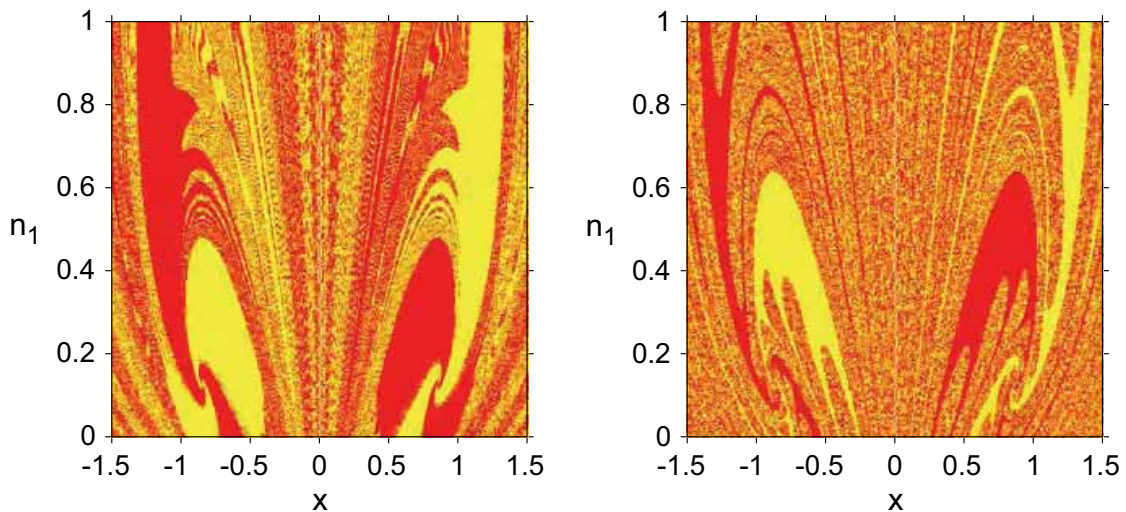
Fig. 9 Basin plots of two co-existing stable 4-cycles, for $\beta = 2.74$ (left) and $\beta = 3$ (right), with other parameters fixed at $a = 0$, $b = 0.5$, $s = 1.35$, $C = 1$, $w = 0$ and $\delta = 0$. The horizontal axis shows the deviation x from the steady state price and the vertical axis the fraction n_1 of rational agents. Different colors correspond to initial states converging to different stable 4-cycles. For each initial state a transient time of 1000 has been used, the resolution is 450×450 and convergence distance $\epsilon = 0.001$. On a Pentium 3 GHz processor, these figures took 13 min and 30 s of computation time each

4.6 Basin of Attraction

When there are multiple, co-existing attractors, the long run dynamical behavior depends on the initial state. The *basin of attraction* of an attractor is the set of initial conditions converging to that attractor. Using the Basin of Attraction menu, different colors are assigned to initial states converging to different attractors. Figure 9 shows an example with two co-existing stable 4-cycles. Red points converge to one stable 4-cycle, while yellow points converge to the other (symmetrically opposite) stable 4-cycle. Figure 9 illustrates that the *basin boundary*, i.e. the boundary between the red and yellow basins of attraction, may have a fractal structure.

In the basin of attraction window, the user can specify the number of attractors (i.e. stable cycles), the distance ε of convergence, the (two) variables for the initial state, the maximum number of transient iterations and the maximum distance of divergence (if exceeded, the point is colored as diverging to infinity). Furthermore, the user needs to specify the period P of each stable cycle and the coordinates of one point on each stable cycle. The latter can be entered either manually or automatically (by clicking on the specified cell the coordinates after a specified number of iterations is pasted automatically). Finally, the range of initial states and the resolution can be specified.

The program starts iterating from a point within the specified range of initial conditions, and iterates it for the specified transient period. Convergence to the specified periodic cycle with period P is assumed to have occurred when the distance from a point within the next P iterations is close enough (within distance ε, specified by the user) to the specified point on this cycle. Divergence (to infinity) is assumed to occur when the distance exceeds a certain pre-specified value of the parameter, referred to as the "Maximum distance". The distance is computed on the basis of the Euclidean norm

using either the two-dimensional space for which the basins are plotted or all dimensions of the model (if the "All dim." box is checked). 'Non Convergence' is reported, when the point has not reached one of the stable cycles within the specified distance after the specified transient. The parameter "Resolution" determines how many points from the range of initial values are used in the algorithm. The default value is set to the maximum screen resolution, but initial explorations with lower resolution (e.g. 100) is recommended, because high resolution basin plots are computationally intensive.

5 Simulations with Noise

There are a few remaining features of *E&F Chaos* we would like to mention. The Basics menu contains two submenus, Spreadsheet and Statistics. The Spreadsheet option is very helpful in saving numerical values of all variables. As all other graphical windows, the Spreadsheet window has an Edit/Parameters submenu where parameters and initial states can be modified. The user can also specify the first point and the number of points to be computed. Using the File/Save as submenu, the data can be saved as a text file. The Basics/Statistics menu allows the user to compute some basic statistics of a time series, such as the mean, median, maximum, minimum, standard deviation, variance, and the skewness and kurtosis coefficients.

An important question, particularly relevant for economics, is how noise affects the dynamics of a nonlinear system. The user can include simulated noise in the model directly in the LUA code using the LUA random number generator `math.random`, which is an interface to the well-documented ANSI C random number generator `rand()`. However, for the convenience of the user, *E&F Chaos* also has a built-in noise option using the Delphi Pascal noise generator, intended for obtaining a quick impression of the effect of additive noise with various distributions on the state variables. Upon clicking the Extra/Noise submenu, the Noise window appears, where the user can select the distribution of the noise (uniform, normal or double exponential), the standard deviation of the noise and the variable to which the noise should be added. The default option is then that in each time step a noise term is added to the selected variable after each iteration. The user has two additional options, to add a noise term only each k-th period or to select a probability p with which the noise is added in each period. In this way, the user can easily investigate how noise affects the nonlinear dynamics.

As an example, Fig. 10 shows the same bifurcation diagram as in Fig. 2, subject to dynamic noise. In each time step, a normally distributed random variable ε_t, with zero mean and standard deviation 0.1 has been added to the RHS of the dynamics of the expected price in Eq. 5. The figure shows that the detailed structure of the bifurcation diagram disappears. However, in the presence of noise price fluctuations exhibit similar features as in the noise free case. For small values of the parameter w, $0 \leq w \leq 0.3$, price fluctuations are close to the steady state, for large values of w, $w > 0.8$, prices exhibit up and down oscillations (a noisy 2-cycle), while for intermediate values of w, $0.3 < w < 0.7$ price fluctuations are complicated. An observant reader may still detect a noisy 4-cycle in the bifurcation diagram in Fig. 10, for $0.7 < w < 0.75$.

Fig. 10 Bifurcation diagram as in Fig. 2, buffeted with noise ϵ_t, normally distributed with SD 0.1. The diagram shows the long run behavior (100 iterations) after a transient of 100 iterations
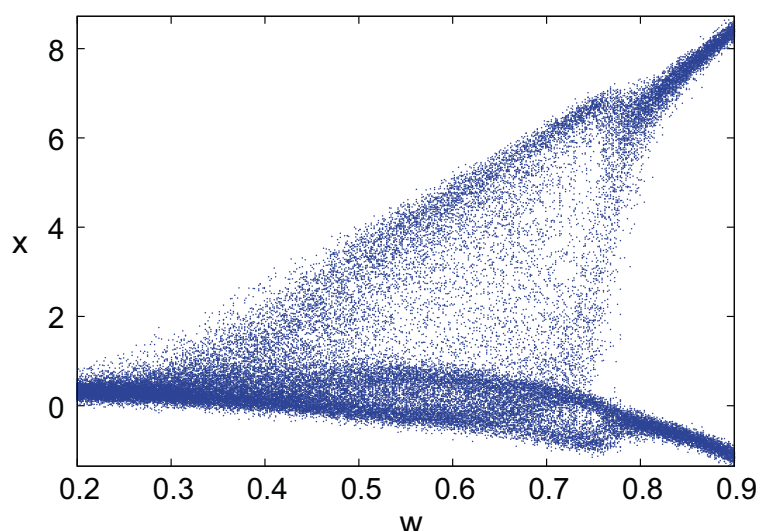
6 Summary

In this paper we have described the *E&F Chaos* software package for simulating and analyzing nonlinear economic dynamical systems. The current version is an upgraded version of an earlier program, the main added features being the ability for the user to easily specify new models in the LUA language without the need of a compiler, as well as the ability to generate publication quality graphical output. The paper illustrates how the different features of the program can be used when the *E&F Chaos* is employed for carrying out its primary task, which is to aid students and researchers in the initial stages with a quick numerical analysis of the dynamical features of newly specified nonlinear models. To this end we have considered two case studies, comprising of both a one-dimensional and a higher-dimensional economic dynamic model, and have described in detail how the *E&F Chaos* program could be utilized for quickly obtaining an overview of the dynamical properties of these models, and how *E&F Chaos* can help formulate more advanced research questions for further analysis. In addition, we briefly discussed the program's built-in feature to investigate how noise affects the nonlinear dynamics. We hope that this software package will contribute to more applications of nonlinear dynamics in economics and finance, and make it easier for non-specialists to discover the intriguing features of nonlinear dynamics.

References

Arrowsmith, D. K., & Place, C. M. (1995). *An introduction to dynamical systems*. Cambridge: Cambridge University Press.

Boldrin, M., & Woodford, M. (1990). Equilibrium models displaying endogenous fluctuations and chaos: A survey. *Journal of Monetary Economics, 25*, 189–222.

Brock, W. A., & Hommes, C. H. (1997). A rational route to randomness. *Econometrica, 65*, 1059–1095.

Brock, W. A., Hsieh, D. A., & LeBaron, B. (1991). *Nonlinear dynamics, chaos and instability: Statistical theory and economic evidence*. Cambridge: MIT Press.

Chiarella, C. (1988). The cobweb model: Its instability and the onset of chaos. *Economic Modeling, 5*, 377–384.

Day, R. H. (1994). *Complex economic dynamics. Volume I: An introduction to dynamical systems and market mechanisms*. Cambridge: MIT Press.

Devaney, R. L. (1989). *An introduction to chaotic dynamical systems* (2nd ed.). Redwood City: Addison Wesley Publication.

Diks, C. G. H., & van der Weide, R. (2005). Herding, a-synchronous updating and heterogeneity in memory in a CBS. *Journal of Economic Dynamics and Control, 29*, 741–763.

Doedel, E. J., Paffenroth, R. C., Champneys, A. R., Fairgrieve, T. F., Kuznetsov, Y. A., Oldeman, B. E., Sandstede, B., & Wang, X. J. (2001). AUTO2000: Continuation and bifurcation software for ordinary differential equations. Applied and Computational Mathematics. California Institute of Technology. http://indy.cs.concordia.ca/auto/.

Ezekiel, M. (1938). The cobweb theorem. *Quarterly Journal of Economics, 52*, 255–280.

Grandmont, J.-M. (1985). On endogenous competitive business cycles. *Econometrica, 53*, 995–1046.

Grandmont, J.-M. (1988). Nonlinear difference equations, bifurcations and chaos: An introduction. CEPREMAP Working Paper No 8811, June 1988.

Guckenheimer, J., & Holmes, P. (1983). *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*. New York: Springer Verlag.

Hommes, C. H. (1991). Chaotic dynamics in economic models. Some simple case-studies. Groningen Theses in Economics, Management & Organization, Wolters-Noordhoff, Groningen.

Hommes, C. H. (1994). Dynamics of the cobweb model with adaptive expectations and nonlinear supply and demand. *Journal of Economic Behavior & Organization, 24*, 315–335.

Hommes, C. H. (2006). Heterogeneous agent models in economics and finance, In L. Tesfatsion & K. L. Judd (eds.), *Handbook of computational economics, volume 2: Agent-based computational economics* (pp. 1109–1186). Amsterdam: North-Holland, Chap. 23.

Hommes, C. H., Huang, H., & Wang, D. (2005). A robust rational route to randomness in a simple asset pricing model. *Journal of Economic Dynamics and Control, 29*, 1043–1072.

Huberman, B. A., & Glance, N. S. (1993). Evolutionary games and computer simulations. *Proceedings of the National Academy of Sciences of the United States of America, 90*, 7716–7718.

Kuznetsov, Y. (1995). *Elements of applied bifurcation theory*. New York: Springer Verlag.

LeBaron, B. (2006), Agent-based computational finance. In L. Tesfatsion & K. L. Judd (eds.), *Handbook of computational economics, volume 2: Agent-based computational economics* (pp. 1187–1233). Amsterdam: North-Holland, Chap. 24.

Li, T. Y., & Yorke, J. A. (1975). Period three implies chaos. *American Mathematical Monthly, 82*, 985–992.

Medio, A. (1992). *Chaotic dynamics. Theory and applications to economics*. Cambridge: Cambridge University Press.

Medio, A., & Lines, M. (2001) *Nonlinear dynamics: A primer*. Cambridge: Cambridge University Press.

Mira, C., Gardini, L., Barugola, A., & Cathala, J.-C. (1996). *Chaotic dynamics in two-dimensional noninvertible maps*. Singapore: World Scientific.

Muth, J. F. (1961). Rational expectations and the theory of price movements. *Econometrica, 29*, 315–335.

Nerlove, M. (1958). Adaptive expectations and cobweb phenomena. *Quarterly Journal of Economics, 72*, 227–240.

Nowak, M., & May, R. M. (1992). Evolutionary games and spatial chaos. *Nature, 359*, 826–929.

Nowak, M., Bonhoeffer, S., & May, R. M. (1992). Spatial and the maintainance of cooperation. *Proceedings of the National Academy of Sciences of the United States of America, 91*, 4877–4881.

Nusse, H. E., & Yorke, J. A. (1998). *Dynamics: Numerical explorations* (2nd ed.). Applied Mathematical Sciences (Vol. 101). Springer-Verlag.

Palis, J., & Takens, F. (1993). *Hyperbolicity and sensitive chaotic dynamics at homoclinic bifurcations*. Cambridge: Cambridge University Press.

Racine, J. (2006). gnuplot 4.0: A portable interactive plotting utility. *Journal of Applied Econometrics, 21*, 133–141.

Rosser, J. B. (2000). From catastrophe to chaos: A general theory of economic discontinuities. Boston: Kluwer.

Wolf, A., Swift, J. B., Swinney, L., & Vastano, J. A. (1985). Determining Lyapunov exponents from a time series. *Physica D, 16*, 285–317.