Agent Based Modeling of Growth Processes

Ralph Abraham*

Abstract

Growth processes abound in nature, and are frequently the target of modeling exercises in the sciences. In this article we illustrate an agent-based approach to modeling, in the case of a single example from the social sciences: bullying..[1]

CONTENTS

*Mathematics Department, University of California, Santa Cruz, CA USA-95064. *rha@ucsc.edu*
[1]Submitted to *Mind, Brain, and Education* .

INTRODUCTION

Growth processes are ubiquitous in nature. In the physical sciences we find them in diffusion aggregation, phase transitions, and wherever physical pattern formation occurs. In biology, we see them at work in the foundations of life itself, in embryogenesis, protein synthesis, and in the activity of living heart muscles, neural cortices, in wound healing, and so on. And in the social sciences we see them at work in the development of social networks and institutions (political, religious, recreational, etc) and in the global economy. Growth, development, and pattern formation are among the synonyms denoting a process similar to the emergence of a photographic image in a darkroom. True, they are not precisely synonyms, as for example, economic growth and economic development have slightly different meanings.

Nevertheless, for all these processes we shall favor the name, growth, as it is short, and was the first choice of Sir D'Arcy Wentworth Thompson (1860 – 1948) for the title of his seminal book, *On Growth and Form*, of 1915, which championed morphogenesis, the process by which patterns are formed in plants and animals. For growth processes of this sort, complex dynamical systems are especially appropriate models, and NetLogo is a very convenient agent-based modeling tool.

We have chosen bullying as a target for agent-based modeling, as it is widely known and discussed theses days, and most readers will understand its meaning without recourse to the wikipedia. Actually, it refers to many similar phenomena, including (to name just a few) racial violence, gender abuse, and gay bashing, in addition to cheating and the violence widely reported among school children. The bully/victim dichotomy is smear to the hawk/dove opinion groups responsible for arms races.[2]

We regard bullying, in whatever context, as a growth process. For example, in the school yard, a class of children may be patternless on the first day of school, but soon a pattern emerges in which bullies and victims are differentiated. It is this process of growth and form that we propose to model. In this we will be aided by the existence of a substantial literature of mathematical models and computer simulations. In particular, the literatures of network theory, complex dynamical systems, and catastrophe theory will be our mathematical starting points.

Complexity is an ambiguous term referring to some of these ideas: chaos theory, cellular automata, genetic algorithms, networks, system dynamics, and complex sys-

[2] Gavrilets

tems.[3] Complex dynamical systems are convenient in this context, as the gradual change of a neutral person to a bully or victim may be accommodated by changing a control parameter in the dynamical scheme at a node, as opposed to having needing types of nodes in the social network.

Although bullying may be regarded as a topic in social psychology in the spirit of Kurt Lewin, our intention is only to illustrate the use of complex dynamical systems in modeling a growth process in whatever field.

Our modeling activity will be formulated in the NetLogo agent-based modeling environment as it has many virtues suitable for the social sciences, including: it is easy to learn, rapid to program, provides drag-and-drop creation of systems of ordinary differential equations as well as discrete dynamical systems, offers excellent graphics in two or three dimensions, music synthesis, animation with video output, and many other virtues.

Part I. COMPLEXITY

Graphs, networks, dynamical systems and schemes, complex dynamical systems – these are all part of the sciences of complexity.[4]

1. Graphs

Sometimes *graph* and *network* are used interchangeably, meaning a structure comprising nodes and directed or undirected links. But here we will treat them as separate structures. By a *graph* we mean a finite set of points in the plane called *vertices*, connected by non-directed line segments called *edges*, and having no more than one edge connecting any two vertices.[5] Graph theory is a relatively new branch of mathematics.

While it may be conventional to give Euler credit for creating graph theory in 1736, we may begin our history with the English mathematician, James Joseph Sylvester, (1814-1897) who contributed the word graph to the literature of mathematics in 1879.

[3] An excellent text is (Mitchell, 2009).
[4] See (Mitchell, 2009).
[5] This is official known as a simple planar graph.

Following contributions by many mathematicians in the early 20th century, the ideas of graph theory were applied to communication networks, especially in the works of the Hungarian mathematician, Paul Erdös (1913-1996).[6] Around 1960, along with Alfréd Rényi, Erdös obtained many results on random graphs, in which numerous vertices are connected at random by edges.

The application of graph ideas to sociology soon followed, notably in the work of social psychologist Stanley Milgram (1933-1984) in 1967 on a property called *six degrees of freedom*, and that of the sociologist Mark Granovetter (b. 1943) in 1973, giving rise to the transdisciplinary field called social network analysis (SNA), or more generally the science of networks [7]

A further quantum jump of SNA history occurred in 1999, when Hungarian physicist Albert-László Barábasi (b. 1967) and his graduate student Réka Albert (b. 1972) discovered the *scale-free (or power law) property* of SNA graphs through a careful study of data from three different types of actual network graphs. This property is observed in a histogram of graph data: a list of how many vertices have a given number of edges.

Suppose for a positive integer, k, the number of vertices of our graph having k edges attached is $N(k)$. Then the graph is scale-free with a power law of *degree exponent* γ, a positive real number, when this formula is satisfied, at least approximately,

$$N(k) = N(0)k^{-\gamma}$$

For example, four scale-free graphs are shown in Figure 1, along with their values of V, E, and γ.

In generating exemplary graphs such as these, it is helpful to keep in mind a formula discovered by Euler back in the 18th century, constraining the numbers of vertices, edges, and faces of a of a convex polyhedron. Euler's formula implies that for a connected graph with V vertices and E edges, $E \leq 3V - 6$.

2. Dynamics and Growth

For graph theory, a graph is static; it is fixed in time, and its properties are studied. However, in applications, graphs may be encountered which are dynamic, that is,

[6] See (Watts, 2003; p. 43.

[7] See (Watts, 2003; p. 50).

their configuration of vertices and edges may change with time. One type of such dynamic behavior is *growth*. This idea of growth is a natural one for the field of complexity. A graph may increase in its number of vertices, perhaps by adding one or a few at a time. When a new vertex is added, edges may be added at once or later. Alternatively, with a fixed number of vertices, edges may appear or disappear.

In studying the statistics of large real graphs, Barábasi deduced a reason for graphs to exhibit a scale-free property, a reason based on the growth of the graph, vertex by vertex and edge by edge. This growth strategy, called *preferential attachment*, consists of a newly appearing vertex preferring to attach an edge to an existing vertex which is popular, that is, has lots of edges already. This growth strategy is a natural one, for example, for the friendships of children in school.

3. Networks

By *network* we shall mean, in the following, a set of points called *nodes*, connected by directed line segments called *links*, which may be endowed attributes (most commonly a number called a weight). It is this type of network which provides the foundation for complex dynamical systems, also known as system dynamics in the pioneering work of Jay W. Forrester. Note that a network overlies a graph, which may be obtained by forgetting the directions and attributes of its links. But a network in general is not simple. That is, two nodes are frequently connected by two links, one directed each way. Thus if we forget the directions, we might have a non-simple graph having two edges connecting two vertices. In this case we could combine the two edges into one, obtaining a simple graph. Let us say a network is simple if for any two nodes, there is at most one link connecting them in each direction. Let us assume this property throughout. Then we may say that underlying every network there is a unique graph; both are simple.

As a typical example of a network, let us consider the World Wide Web (WWW). Each webpage or resource belongs to a website which is hosted on a machine with a unique IP address. Each host machine may be regarded as a node of the internet, which is a very complex network of host machines, routers, and route algorithms. But each webpage has a unique Uniform Resource Locator (URL). These URLs are the addresses for one resource to link to another. So the WWW may be regarded as a network of resources (nodes) linked by anchors which move a browser from one resource to another (links). Further, the links may or may not be characterized by attributes.

Regarding the of Barábasi and Albert (1999) on the scale-free (power-law) property of the WWW, the basis of their analysis was to count separately the number of incoming and outgoing links for each node. Looking at the two histograms (in log-log plots) revealed two separate power laws: exponent 2.1 for incoming links, and 2.5 for outgoing links.[8]

The WWW began with a few nodes n 1993, and today comprises at least 3.65 billion.[9] Clearly it epitomizes the word growth, as far as network nodes are concerned. An adult human brain may have 85 billion neurons, and 10^{14} synapses. A neural network epitomizes growth in another way. Although in the early development of an animal the number of neurons (nodes) is growing, eventually the mature brain continues growing through the modification of synaptic connections (links) and their weights (attributes) in the process called *plasticity*. This type of growth is responsible for *learning*, the emergence of new behaviors in a large network, especially a *social network*.

In an artificial neural network, a mathematical model, a neuron is not only a network node, it is also a dynamical scheme, a mathematical structure to which we now turn.

4. Dynamical systems and schemes

Dynamical systems theory, or DST, also known as nonlinear dynamics, and as the qualitative theory of systems of ordinary differential equations, is a new branch of mathematics originating with Poincaré around 1880. Dynamical systems occur in three types, all due to Poincaré:[10]

- a *flow*, defined by a vectorfield (a system of autonomous ordinary differential equations) defined on a state space, S,
- a *cascade*, defined by a reversible mapping of S onto itself, or
- an *iteration*, defined by an arbitrary mapping of S into itself.

The state space, S, is in general a finite-dimensional differentiable manifold, but here we will consider only the special case, $S \subset R^n$ (cartesian n-space). Also, here we will consider only dynamical systems of flow type.

[8] (Barábasi, 2002; p. 68, fn 2)
[9] August 27, 2013, http://www.worldwidewebsize.com.
[10] See (Abraham, 1993; sec. 2).

We will need a basic vocabulary of dynamical systems theory.[11] In this context, each point of S series as the initial point of a unique curve in S, parameterized by the time variable, t, and extending into both future and past times, until either running off the state space, or approaching asymptotically to a non-empty limit set, L. The *inset* of a future limit set, $In(L)$, consists of every initial point having L as its future limit set. In the case in which $In(L)$ is an open set of S, then L is called an *attractor*, and the open set $In(L)$ is called the *basin*, or *basin of attraction* of L. The attractors of flows occur in three varieties:

- a static attractors, which is a single point,
- a periodic attractor, which is a closed loop, and
- a chaotic attractor, which means anything else.

The basic understanding of a flow, according to DST, is provided by a map of S showing the distribution of basins of attraction. We call this the *attractor-basin portrait*, or *AB-portrait*, of the system. Each basin contains a single attractor, and in the general case, almost every point of S belongs to one of the basins. The exceptional points comprise the boundaries, or *separatrices*, of the basins.

This is minimum vocabulary for an appreciation of dynamical systems. But now we must go on to dynamical schemes.

A *dynamical scheme* is a dynamical system depending on parameters. These parameters are sometimes called *control variables*. This is due to the many applications of DST in which a model for an experimental setup has states which change according to dynamical rules, which in turn depend upon control knobs in the hands of the experimentalist. For each setting of the controls, the scheme specifies a unique flow, along with its AB-portrait. A major change in the AB-portrait caused by a small change in the controls is known as a *bifurcation*. The bifurcations of dynamical schemes, which are of cardinal importance in the application of DST, occur in three varieties:

- a subtle bifurcation, in which an attractor changes type in a small way,
- a catastrophic bifurcation, when an attractor and its basin appears or disappears, and
- an explosive bifurcation, affecting a sudden change in the size of an attractor.

We are going to approach an understanding of schemes and their bifurcations through a few simple examples.

[11] See (Abraham and Shaw, 2005).

5. Exemplary bifurcations

We will consider several exemplary bifurcations, each in a different dynamical scheme, to exhibit all three ties of bifurcation.

- a *Hopf bifurcation*, a subtle bifurcation with one control, a static attractor changes into a periodic attractor,
- a *fold catastrophe*, a catastrophic bifurcation with one control, a point attractor and its basin appears (or disappears),
- a *cusp catastrophe*, with two controls, combining catastrophic and subtle features,
- a *blue loop bifurcation*, an explosive bifurcation with one control, a point attractor explodes into a periodic attractor, and
- a *double cusp catastrophe*, a complex event with eight controls, originally developed to model arms races.

5.1 Hopf bifurcation

Our first exemplary scheme was created by Paul van Geert and Henderien Steenbeek to model the control of aggressive behavior in a classroom.[12] It exhibits a subtle bifurcation in a two-dimensional scheme with state variables X and Y, where X is in the range $(0, 0.5)$ and represents the level of misbehavior in the classroom, and Y is in the range $(0, 2)$ and is the level of punishment chosen by the teacher to control the situation. The state space is an open rectangle. The control space is an open interval of the real numbers. The single control parameter, c, varies in the range $(0, 10)$, representing the effect of misbehavior on punishment. Each value of the control parameter determines a unique flow, generated by the vectorfield,

$$\begin{aligned} X' &= X(1-X) - aXY \\ Y' &= -bY + cXY \end{aligned}$$

where $a = 1.45$ and $b = 0.6$, and c may vary. This is our dynamical scheme.

With lower values of the control, the flow has a single attractor, and it is static. Nearby trajectories approach it by spiraling in. With higher values, there is a single attractor, but it is periodic. If the control is slowly changed from zero to ten, there will be a subtle change at a value near $c = 7.5$, at which value the attractor changes

[12]van Geert and Steenbeek)

from a point to a very small loop, indicating an oscillation in the values of X and Y with a period around 10 units of time, but a very small amplitude. As the control is further increased, the amplitude of the oscillation increases, and the point (X, Y) may be seen to move around a loop in the two-dimensional state space. As the control is increased yet further, the amplitude of the oscillation continues to increase.

The response diagram of the Hopf bifurcation is shown in Figure 2.[13] This represents an identical copy of the state space (an open rectangle) for each and every point in the control space (horizontal open interval). The plane rectangle on the left is the state space of the scheme, showing the AB portrait of the flow with $c = 0$. A trajectory (blue curve) spirals in, approaching the point attractor (red point) asymptotically.

The plane rectangle on the right is the state space again, showing a trajectory (blue curve) spiraling out from a repeller (green point) at the origin in the center, and spiraling in to the periodic attractor (red loop) from the inside. Other trajectories (not shown) are spiraling in to the periodic attractor from the outside.

The plane rectangle in the center shows the portrait of the flow at the moment of bifurcation, at which the central point changes from attracting (red) to repelling (green).

5.2 Fold bifurcations

We will consider two exemplary schemes exhibiting catastrophic bifurcations.

Our second exemplary scheme comes from *elementary catastrophe theory*, where it was discovered by René Thom in the 1960s as an archetypal bifurcation based on ideas from differential topology. It exhibits a catastrophic bifurcation in a one-dimensional scheme with state variable X and a single control parameter, C. On the state space there is a vector field, mostly pointing downward. As the control parameter increases, there is a gradual change introduced in the vectorfield, but a discontinuous change occurs in its AB-portrait. This is our dynamical scheme for the fold bifurcation.

The response diagram of this bifurcation is shown in Figure 3.[14] FIGURE 3 The vertical line segment represents the state space of the scheme; five copies are shown in

[13]This is figure 17.1.7 from (Abraham and Shaw, 2005).
[14]This is figure 18.1.8 from (Abraham and Shaw, 2005).

the figure. Reading from right to left, we see first an AB-portrait with one attractor (red) and one repeller (green). The repeller is the lower boundary of the basin of the attractor. Points above the attractor move downward, while points between the repeller and the attractor move upward.

As the control parameter is decreased to the left, the attractor and the repeller move closer together; the attractor is approaching the boundary of its basin of attraction.

In the center, the attractor and the repeller meet, and each disappears. This is the bifurcation point. For controls to the left of the bifurcation point, all motion ss downward, there is no attractor in view.

Imagine an initial point in the upper right of our response diagram. Following its trajectory without changing the control parameter, it moves rapidly downward, slowing and coming to rest just a bit above the attractor. Now imagine that the control parameter is slowly moved to the left. Our trajectory tracks obediently along with the attractor as long as it can. But as soon as the control parameter passes the bifurcation value, our moving point finds itself in another basin, belong apparently to an attractor far below. Its trajectory rushes off downward, accelerating madly. This is a typical catastrophe.

This scheme and response diagram is abstract, and in an actual model we are more likely to find a response diagram with more than one bifurcation. Consider now the double-fold diagram, shown in Figure 4. This is composed of two folds put together. As in the fold described above, the state space is a vertical line segment, and the control space is a horizontal line segment. The locus of attraction (red) has two branches, while the locus of repulsion (green) has one branch.

Beginning as before with a control value at the right end of the diagram, there is a single basin containing a point attractor. Moving the control slowly to the left, our trajectory will track the locus of attraction (red) until it disappears in a fold catastrophe. Then the point we are following finds itself in the basin of another attractor, below, and off it zooms, until it is caught by the lower branch of the locus of attraction. It has experienced a catastrophic bifurcation, a fast transient between two slower motions.

Now if we try to retrace the event by moving the carol slowly to the left, our point of interest will track the lower attractor as far as it can, then zoom back up to its original location near the upper attractor, in a second catastrophic bifurcation. But the up-jump as far from the down-jump. This is a prototypical hysteresis loop.

5.3 Cusp catastrophe

Our third example, also from elementary catastrophe theory, has again an open line segment as state space, with variable x, but the control space is two dimensional, of points (a, b), as shown in Figure 5. Here x, the state variable, is vertical, while the plane of (a, b) is horizontal. The control parameter a is called the *normal factor*, while b is called the *splitting factor*. The folded sheet (with the heavy black outline) is called the *cusp surface*.

Consider a vertical plane that cuts through Figure 5 parallel to the (a, x) plane, with its front-back position determined by a fixed value of b, the splitting factor. When b is at the from of the figure, the cutting plane cuts through the cusp surface in the double-fold curve shown in Figure 5. This is a response diagram with two bifurcation points, both fold catastrophes.

But when the cutting plane is determined by a smaller value of b, the cusp surface is cut in the red curve shown in Figure 6. This is the response diagram of a scheme with no bifurcation at all. As the control parameter a is moved, our attractor moves about, following the locus of attraction shown in green in Figure 6.

Thus, we may regard the splitting factor, b, as the creator of a double fold, which occurs subtly. Thus, the cusp catastrophe combines both subtle and catastrophic features, but is called a catastrophe as all of the bifurcations occurring within it are catastrophic. Also, it is a key figure in elementary catastrophe theory, as originally described by René Thom.

This scheme has many applications, the early examples being treated in detail in the first literature on catastrophe theory.[15] One of these early examples is a model of opinion formation in groups due to C. A. Isnard and Christopher Zeeman in 1976.[16]

5.4 Blue loop bifurcation

This type of bifurcation was originally identified by Stephen Smale in 1967. We will describe now a simple example published by Christopher Zeeman in 1982.[17] In this scheme the state space is a plane rectangle, several copies of which appear as

[15] See (Zeeman, 1977) and references therein.
[16] See (Zeeman, 1977; chs. 1 and 10) and references therein.
[17] See (Abraham and Shaw, 2005; ch. 21) and references therein.

parallel vertical sheets in Figure 7. The control space is an open line segment, shown horizontally in the figure.

In the first sheet on the left in the figure, the flow has a point attractor (red dot), a point repeller (green dot), and a saddle point (shown half-green and half-red). The saddle point has a one-dimensional inset curve (green) and a one-dimensional outset curve (blue).

In the last sheet on the right in the figure, the flow has a periodic attractor (red) and a point repeller (green).

In the center sheet, the flow has a bifurcation in which the saddle point and the point attractor have collided, and the outset of the saddle has become a periodic attractor. This is the explosion event, in which it appears that a small attractor has suddenly become large.

5.5 Double cusp system

This is a complex dynamical system in which two cusp schemes have been combined, each linked to the other. It was applied to anorexia nevosa by J. Callahan in 1982, and proposed as an arms race model in the tradition of Lewis Fry Richardson (1919) by M. N. Kadyrov in 1984.[18] Here we will examine the geometry of the bifurcation set of this scheme, and return to the applications in a Part II.

We consider two cusp schemes, as described above. Let us denote the states and controls of our first cusp scheme as x and (a, b), respectively, as above, and for the second scheme, y and (c, d).[19] We now couple the two cusp schemes mutually, letting the normal factor of each be proportional to the state of the other. Thus, we replace the first normal factor a with Ay, leaving the first splitting factor, b as a free control parameter of the combined scheme. The constant or proportionality, A, is also a control parameter of the combined scheme. Similarly, we replace the second normal factor c with Cx, leaving the second splitting factor, c as a free control parameter of the combined scheme. The constant or proportionality, C, is also a control parameter of the combined scheme.The combined scheme is a dynamical scheme with two state variables, (x, y), and four control parameters, (A, b, C, d).

The bifurcation set of this scheme has been studied with extensive computations,

[18]See (Abraham, 1986), (Abraham et al, 1991), and references therein.
[19]See (Abraham et al, 1991; p. 419).

in the reduced scheme with two controls, the splitting factors (b, d), obtained by choosing fixed values for the two constants of proportionality, (A, C). In fact, let $A = C = 1$. We call this scheme with two state variables and two control parameters the *Kadyrov scheme*. As shown in Figure 8, its bifurcation set comprises two cusp curves and two fold curves, dividing the control plane into the seven regions A through G.[20] In regime A there are 4, attractors, all static. In each of the regimes B, C, D, and E there are two attractors, both static. In both F and G there is a single periodic attractor, called the *Kadyrov oscillation*.[21]

We go on now to the subject of complex dynamical systems in the abstract, which is fundamental to our modeling strategy for growth or development of structures in a social psychology framework in Part II.

6. Complex dynamical systems

A *complex dynamical system* comprises a directed network together with data: every node contains a *dynamical scheme* (a dynamical system with control parameters), and every directed link connects states at its tail to controls at its head.

The basic ideas of complexity go back to the early years of nonlinear dynamics, in the works on self-oscillation by Lord Rayleigh (1842-1919), Henri Poincaré (1854-1912), Georg Duffing (1861-1944), and Balthasar van der Pol (1889-1959).

They flourished in the interaction of three twentieth century movements: cybernetics, general systems theory, and systems dynamics.[22]

Cybernetics

Cybernetics emerged in the Macy conferences (1946-1953) – organized explicitly to apply these new ideas of mathematics, physics, and computer science to the social sciences. The participants, known as the Macy Core Group, included: Gregory Bateson (1904-1980), anthropologist; Kurt Lewin (1890-1947), among the founders of

[20] Due to Kadyrov. See (Abraham, 1990; Fig. 2) and (Abraham et al, 1991; Fig. 2).

[21] Full details for the various bifurcations indicated in Figure 8 may be found in (Abraham, 1990) and (Abraham et al, 1991).

[22] See (Abraham, 2011) for this story.

social psychology; Warren McCulloch (1898-1969), neurophysiologist (chair); Margaret Mead (1901-1978), anthropologist; Walter Pitts (1923-1969), mathematician and logician; and Norbert Wiener (1894-1964), mathematician and founder of cybernetics; among others. This new way of thinking was the subject of the influential book, *Cybernetics*, of Norbert Wiener in 1948.

Neural networks

The application of the mathematics of self-oscillation to *artificial neural networks* (ANNs) arose in the works of McCulloch and Pitts (1943), following the computer revolution, and were crucial to the emergence of cybernetics in the Macy meetings. Since then, ANN has co-evolved with computer science, incorporating ideas such as Hebbian learning from the study of biological neural networks (BNNs). ANNs today are ubiquitous throughout technology. Through learning algorithms, an ANN modifies the strength of its connections, giving rise to new and useful behaviors. The idea that the intelligence of an ANN (or BNN) lies in this matrix of its connections, rather than the cleverness of its neurons, is known as *connectionism*.

General systems theory

General systems theory, the brainchild of Ludwig von Bertalanffy (1901-1972) in Vienna evolving since the 1920s, may be regarded as a European counterpart to the American cybernetics movement. It has developed outside the mainstream of academia, and spans all the usual fields of universities. Its mathematical branch is called system dynamics.

System dynamics

System dynamics was created by Jay Wright Forrester (b. 1918) in the context of his work with the M.I.T. School of Industrial Management (now the Sloan School of Management). Following brief announcements in the *Harvard Business Review* of 1958 and 1959 came his foundational book *Industrial Dynamics* of 1961, in which a flow chart approach to mathematical modeling of an industrial (or economic) system is introduced. Forrester gives credit for his approach to management to:

- the Nebraska cattle ranch on which he grew up,

- his college education in electrical engineering at the University of Nebaska,
- his graduate education in the M.I.T. Servomechanisms Laboratory,
- his creation of *Whirlwind I*, one of the first high-speed digital computers in the 1940s, for which he invented the magnetic core memory,
- his experience as head of the Digital Computer Division of the M.I.T. Lincoln Laboratory from its beginning in 1951, and
- his work in managing the creation of the Semi-Automatic Ground Environment (SAGE) Air Defense System for the protection of the continental United States.

An appendix to the book introduces *DYNAMO*, a computer programming language created for the modeling and simulation of the flow charts that give visual representation of the systems of ordinary differential equations that underly the flow charts. The first version of DYNAMO was created by Richard Bennett, beginning in 1955, and used in Forrester's 1958 paper. Later versions were used for the models of the world environment on which the 1972 *Limits to Growth* book was based, showing that the human population explosion would eventually destroy the biosphere. The books of Jay Forrester might be a good starting point for anyone aspiring to the mathematical modeling and computer simulation of a growth process.

Eventually, the DYNAMO system evolved into STELLA, a drag-and-drop environment for describing a complex model visually and running simulations. This is still used today, even for high school courses on system dynamics, and even in middle schools. A simplified version of this software is included as a accessory tool in NetLogo, a freeware programming environment for agent-baed modeling, to which we now turn.

Part II. MODELING GROWTH

We have encountered growth ideas for graphs, networks, and complex dynamical systems (CDSs). In a neural network, an important type of CDS, growth (that is, development or learning) takes place by adjustments to the strengths of the directed connecting links. In a social network context, this process might result in the formation of patterns such as opinion groups, political parties, or unwanted behaviors such as racism, gender discrimination, and so on. The behavior modification (or BM) model of van Geert and Steenbeek is just such a process, and has been described by them as a dynamic growth model.[23]

[23] See (van Geert and Steenbeek, 2004; p. 1).

In Part II we would like to build on this model as an exemplary guide to modeling a growth process with CDS ideas, using NetLogo as our modeling environment. From the BM model and the double-cusp model described in Part I we will then construct a CDS model for bullying in a school setting.

7. NetLogo

In the history of computer languages, there are quantum leaps and plateaus. For example, the C language from the early days of the Unix operating system was typical of many languages. If you learned one, you could easily learn all the others.

Suddenly a new concept emerged, the object. An object is a named code package or container with its own addressing scheme. It might contain variables, data, functions, and so on. Then object oriented programming (OOP) languages such as C++ came along. C++ became another language to learn, but it required a cognitive leap. If you made this leap, you could then easily learn also the other OOP languages.

A further jump then followed, agent-based modeling. An agent is (more-or-less) an object that can move around, in a geometrical space or lattice, or on a graph or network.[24].

NetLogo is a free agent-based modeling environment devised by Uri Wilensky, now at Northwestern University. It developed from the Logo programming language created by Seymour Papert and coworkers at MIT around 1967 to teach computer programming concepts to children. Its chief feature was a turtle that could be moved around a plane rectangle through commands such as turn-right, go-forward, and so on. By dropping a pen, the turtle could draw on the plane rectangle. The turtle was the prototypical agent.

In NetLogo as it exists today (2D version 5.0) there are four types of agents: the observer, patches (small rectangles which fill a plane rectangle in a regular lattice), turtles (which move about like Logo turtles), and links (directed or non-directed).

NetLogo programs, which are called models, are built and used for teaching or research in its environment, which comprises:

- a User Interface, which may contain widgets (such as buttons, monitors, plots, sliders, labels, etc), and a Graphics Window (in which the turtles may be be

[24]For additional history, see the introduction in (Abraham, Friedman,and Viotti, 2013)

seen moving about),
- an information page, providing instructions for understanding and using the model, and
- a code section, in which may be written all the code: data, variables, and procedures.

Each agent may have its own data, variables and procedures. For example, turtles (which may belong to different breeds) come with attributes (such as shape, size, color, heading, position, and pen-state) and procedures (such as move, turn, change an attribute, and the like).

The NetLogo system is replete with user manual, code dictionary, tutorials, and exemplary models. It has been designed to be easy to learn, and many scientists and students (down to middle school) have been able to achieve a high level of proficiency in a short time.

Under "Tools" in the top menu bar of the NetLogo environment may be found "System Dynamics Modeler" which will pop up a separate window, in which a STELLA work-alike drag-and-drop system is provided.

This makes it rather easy to write a system of ordinary differential equations using only a few math symbols. The basic icons are stocks (rectangles), flows (heavy directed links containing a valve and connecting stocks), variables (diamonds), and links (thin directed lines connecting variables and valves).

Warnings. The word *link* has two meanings. In the context of networks (in network theory in general and in NetLogo) it is an edge between nodes, but in the context of the NetLogo SD modeling tool, it is a connection between icons. Similarly, the work *flow* means a system of ordinary differential equations in dynamical systems theory, but means a single such equation in NetLogo SD. These ambiguities have bedeviled CDS students for years, but cannot be helped.

This iconic language due to Forrester is general enough to represent an arbitrary CDS. One must be aware that the word *flow* has two meanings here: one in CDS, another in NetLogo. The main restriction is that in NetLogo, stocks must be one-dimensional. A dynamical scheme occupying a node of a CDS may be represented in this iconic language by a diagram comprising several stocks, flows, and variables, perhaps supported by bits of code in the code section of the NetLogo environment.

With this brief introduction to NetLogo, we may now explain implementations of the BM model in NetLogo.

8. The BM model in NetLogo

We are going to use the BM model to demonstrate two ways to model a CDS in NetLogo: as the iteration of a plane endomorphism, (a mapping from an open plane rectangle into itself), as a flow entirely in NetLogo code, and as a flow in the SD modeling tool.

Iteration

We will first consider the BM model as originally presented by the authors, but change the names of the variables. This model was described as an iterated map. For the author's state variables, (m, p) for misbehavior and punishment, we will write (X, Y) as in subsection 5.1 above. The domain of the map is the open rectangle n the plane defined by,

$$X \in (0, 0.5), Y \in (0, 2)$$

Then the map from a point (X, Y) to its image point, $(Xnew, Ynew)$ is defined by the equations,

$$\begin{aligned} Xnew &= X + GX(1 - X) - EXY] \\ Ynew &= Y - TY + FXY] \end{aligned}$$

where the constants are,

- $G = 0.165$, the growth rate of misbehavior,
- $E = 0.24$, the effect of punishment on misbehavior,
- $T = 0.1$, the teacher's aversion to punishing,
- $F = 1.24$, the effect of misbehavior on punishment

Flow, code

The equations for the iterated map as given by the authors just above may be seen to be the Euler method (with $dt = 1.0$) applied to the integration of the vectorfield defined in subsection 5.1 above,

$$\begin{aligned} X' &= X(1 - X) - aXY \\ Y' &= -bY + cXY \end{aligned}$$

where (approximately) $a = 1.45$, $b = 0.6$, and $c = 7.5$. The Euler method approximates a trajectory with a polygon.[25]

By considering the BM model as a flow defined by a vectorfield rather than a discrete dynamic system defined by the iteration of a map, we have access to a model from the NetLogo Models Library for vectorfields. Modication of this model for the BM model requires some expert programming, but may result in a rather sophisticated experimental system. But a much easier path to a working model is provided by the System Dynamics Modeling tool that is bundled with NetLogo, as we now show.

Iteration, SD tool

The NetLogo SD tool is intended for the simulation of a flow by the Euler method with a time step dt that may be set in the SD interface. But by setting $dt = 1.0$ we may use the tool for simulating the iteration of a map. The system described in the iconic language of Forrester is created by drag-and drop, and is very simple, as shown in Figure 9.

Note. This figure shows that the BM model is a complex dynamical system. Its underlaying network has two nodes, mutually linked. The M node has a dynamical scheme comprising the m stock, the $m - growth$ flow, and the $m - growth - rate$ variable. The P node, similarly, has a dynamical scheme comprising three icons. Two links are internal to the schemes, while two other links are directed edges connecting the p stock to the $m - growth - rate$ and the m stock to the $p - growth - rate$.

The NetLogo User Interface for this model is shown in Figure 10. Note that the Graphics Window is not used in this model. Instead, the behavior of the model during a simulation is shown in the two graph plots. The upper plot shows the time sequence of values of X and Y, while the lower plot shows the trajectory of (X, Y).

The NetLogo code for this model, copied from the code panel of the model, is only this:

```
;;; start of code

;;; VARIABLES
globals [
```

[25] See (Abraham and Shaw, 2005; panel 1.2.10).

```
  G Ep T Em ;;; coeffs in mdot, pdot
  mdot pdot ;;; rates
  ]

;;; PROCEDURES

to setup
  ca
  set m 0.2
  set p 0.6
  set G 0.165
  set Ep 0.24
  set T 0.1
  set Em 1.24
  system-dynamics-setup
  set-current-plot "stocks"
  system-dynamics-do-plot
end

to step
  set mdot mrate m p
  set pdot prate m p
  system-dynamics-go
  set-current-plot "stocks"
  system-dynamics-do-plot
  set-current-plot "state space"
  plotxy m p
end

to go
  step
  tick
end

to clear
  clear-all-plots
  reset-ticks
end
```

```
to-report mrate [ u v ]
  report temp-u G * u * ( 1 - u ) - Ep * u * v ;;; from BM-02.txt
end

to-report prate [ u v ]
  report temp-v Em * u * v - T * v
end

;;; end of code
```

The equations defining the map are the last few lines of code. It is that simple.

9. The double cusp model in NetLogo

Our next target for NetLogo modeling with the SD tool is the double cusp scheme. First we need a model for one cusp, then we can couple two copies, similar to the SD diagram for the BM model shown in Figure 9.

One cusp

The cusp is a flow scheme with a one-dimensional state space, and a two-dimensional control space. The first application described by Christopher Zeeman – in a *Scientific American* article of April, 1976 – is to the outbreak of aggression in dogs. Based on a book of Konrad Lorenz, the control parameters are rage and fear, indicted by the openness of the dog's mouth and the layback of it's ears. The state variable increases from flight to fight.

In Zeeman's second application, to humans, the controls are frustration and anxiety, and the state variable increase from self-pity to anger. And in the third, to nations, the controls are cost and fear, and the state variable indicates the war policy, increasing from dove to hawk. This is the arms race model which generated a rash of research.[26] Now back to the mathematics of the model.

Let x denote the state variable, and (a, b) the controls.We may restrict the domain to $(-1, 1)$ for each: x, a, and b. The vectorfield of this flow scheme is defined by the

[26]See also an article of C. A. Isnard and E. C. Zeeman of 1976 (Zeeman, 1976; ch. 10).

cubic polynomial,

$$x' = a + bx - x^3$$

This is the negradient (negative derivative) of the potential function,[27]

$$F = \frac{1}{4}x^4 - ax - \frac{1}{2}bx^2$$

For given controls (a, b) this will have either one, two, or three zeros, depending on the value of the discriminant, $D = 4b^3 - 27a^2$. The catastrophe surface shown in Figure 5 is the locus of zeros of the vectorfield, and the cusp curve in the (a, b) plane is the bifurcation set defined by $D = 0$. For (a, b) inside the cusp, there are three zeros: an attractor, then a repellor, and then another attractor. And for (a, b) outside the cusp, there is only one, an attractor. For example, with $a = 0$ and $b = 1$, (a, b) is inside the cusp, $D = 4$, greater than zero, and there are three zeros of the vectorfield, at $x = -1$, 0, and 1, respectively, attractor, repeller, and attractor. Two screens of a NetLogo model for the cusp scheme n Figures 11 and 12. The cusp surface, from Isnard and Zeeman, is shown in Figure 13.

Two cusps

The arms race model with one cusp describes the partition of the voting population of a nation into hawks and doves, depending on the control parameters of perceived cost (estimation of the cost of a war in terms of finances, destruction, mortality, and so on), b, and threat (fear of defeat), a. Therefore, to model a war involving two nations, we must combine two cusps, one for each nation. Then we would have, for the first nation, control parameters, (a, b), for threat and cost, respectively, and state variable, x, for readiness (armaments) to go to war, and similarly for the second nation, (c, d) and y. The coupling links between the nations would link the state (armaments) of each nation to the threat parameter of the other, with some scaling factor.[28] For example, let us write,

$$\begin{aligned} a &= Ay \\ c &= Cx \end{aligned}$$

This will leave free, as control parameters of the combined system, the two threat parameters, (b, d). Thus the double cusp model for two nations has two state variables

[27] See (Zeeman, 1977; p. 27).

[28] Here we follow (Abraham et al, 1991).

(x, y), and two control parameters, (b, d). The screens of a NetLogo model for this complex system are shown in Figures 14 and 15.

One could envision the extension of this model for two nations to a more complex system involving a network of nations. Such a model has been called a *cuspoidal network*.[29] Zeeman has devised a clever visualization strategy for these networks in his work modeling the heartbeat.[30] As the instantaneous state at each node defines a unique point, (a, b, x), in the response diagram, the combined states at all nodes of the network may be seen as a cloud of such points, roving over the locus of attraction, the cusp surface.

10. Bullying

In the extensive literature on the problem of bullying in schools, there are various distinctions of roles played by participants. For example, bullies, victims, bully enablers, victim enablers, and pacifiers, have been studied. Sergey Gavrilets has made a case for an evolutionary advantage of pacifiers.[31] Following this idea, I propose here to describe a simple model for a network of three groups: bullies, victims, and pacifiers. In this modeling exercise, intended as an example of CDS thinking, we will combine a double cusp scheme for the bully and victim groups, and a behavior modification scheme for the role of the pacification group.

We may begin by imagining a situation in which an initially homogenous group of strangers evolves a network structure through a process of preferential attachment (see Section 2 above.) Let the psychological attributes of threat, cost, and aggression of each be represented in the cusp model.

As the strong get stronger and weak get weaker, the representations of each in the cusp diagram begin to cluster around the upper or lower sheets of the cusp surface. This growth phenomenon precedes the construction of our model, and a network diagram with two clusters may be assumed: bullies and victims. Let the average behavior of each be modeled with it own cusp scheme, and the interaction be modeled by the links of the double cusp scheme.

The sum (or average) of the states, x and y, may be taken as the level of misbehavior,

[29] See (Abraham,1990).
[30] See Figure 20 in (Zeeman, 1977; ch. 3).
[31] (Gavrilets, 2012)

m, by the pacifiers. Following the indications of the BM model, the pacifiers decide on a level of punishment, p, to assert. This is effected by adding to the cost of each group, bullies and victims.

Ignoring the details of an actual NetLogo SD model for this CDS, let us observe that the double cusp scheme is a complex system with two nodes, bullies and victims, while the BM system is similarly a scheme with two nodes, for misbehavior and punishment. So the combined system described qualitatively here is then a complex scheme with four nodes, combining two similar pairs.

Recalling that the double cusp model has a periodic attractor, the Kadyrov oscillation, for some regions in its control plane, while the BM model has a Hopf bifurcation, it is possible that a forced oscillator situation, and thus chaotic behavior, may occur in our combined model.

CONCLUSION

In Part I we have provided a short course of the basic ideas of complex dynamical system (CDS) theory, and in Part II an exemplary construction of CDS model for a growth phenomenon. While the treatment in each part may be too concise to enable a beginner at mathematical modeling and computer simulation to actually build a CDS model, I believe that a path has been indicated that, with the references, may lead one to mastery of the arcane art of CDS modeling, and an advance level of systems thinking.

ACKNOWLEDGEMENTS

My thanks to my editors, Kurt Fischer and Todd Rose, for instigation, my brother Fred Abraham and partner Ray Gwyn Smith for ideas, my coauthor, Chris Shaw for his drawings, my colleagues Chris Zeeman, Yoshisuke Ueda, Bruce Stewart, and Paul van Geert for helpful communications.

REFERENCES

Books

Abraham, Ralph H., and Christopher D. Shaw (2005). *Dynamics, the Geometry of Behavior, 4th edn.* Santa Cruz, CA: Aerial Press.

Barábasi, Albert-László (2002). *Linked: How Everything Is Connected to Everything Else and What it Means for Business, Science, and Everyday Life.* New York: Penguin.

Mitchell, Melanie (2009). *Complexity, A Guided Tour.* Oxford: Oxford University Press.

van Geert, Paul L. C. (1994). *Dynamic Systems of Development: Change between Complexity and Chaos.* London: Harvester Wheatsheaf.

Watt, Duncan (2003). *Six Degrees: The Science of a Connected Age.* New York, NY: Norton.

Zeeman, E. Christopher (1977). *Catastrophe Theory: Selected Papers, 1972 – 1977.* Reading, MA: Addison-Wesley.

Articles

Abraham, Ralph H. (1984). Complex dynamical systems. In: *Mathematical Modelling in Science and Technology*, Xavier J,R Avula, Rudolf E. Kalman, Anthanasios I. Liapis, Ervin Y. Rodin (Eds.), Pergamon Press; pp. 82 - 86.

Abraham, Ralph H. (1986). Mathematics and evolution: a manifesto. *IS Journal*, 3; pp. 14 - 27.

Abraham, Ralph H. (1987a). Dynamics and self-organization. In: *Self-Organizing Systems*, F. Eugene Yates (ed.). Plenum, 1987; pp. 599 - 613.

Abraham, Ralph H. (1987b). Complex dynamics and the social sciences. *World Futures* 23, 1-10.

Abraham, Ralph H. (1990). Cuspoidal nets. In: *Toward a Just Society for Future Generations, Vol. II*, Int'l. Society for the Systems Sciences , B. A. and B. H. Banathy, eds., pp. 667-683.

Abraham, Ralph H. (1993). Economics and the environment, global erodynamic models. In: *Nonlinear Dynamics in Economics and Social Sciences*. F. Gori, L. Geronazzo, and M. Galeotti (eds.), Berlin: Springer-Verlag; pp. 3-16.

Abraham, Ralph H. (2011). The genesis of complexity. *World Futures*, 67; pp. 380-394.

Abraham, Ralph H., and Christopher D. Shaw (1987). Dynamics: A Visual Introduction. In: *Self-Organizing Systems*, F. Eugene Yates (ed.). Plenum, 1987; pp. 543 - 597.

Abraham, Ralph H., Gottfried Mayer-Kress, Alex Keith, and Matthew Koebbe (1991). Computational unfolding of double-cusp models of opinion formation. *Int. J. Bifurcation and Chaos*, 1(2): pp. 417 - 430.

Abraham, Ralph H., Dan Friedman and Paul Viotti (2013), Complex Dynamical Systems and the Social Sciences. In: *Qualitative Mathematics for the Social Sciences: Mathematical Models for Research on Cultural Dynamics*, Lee Rudolph, (Ed.); pp. 321-341. New York, NY: Routledge.

Gavrilets, Sergey (2012). On the evolutionary origins of the egalitarian syndrome. *Proc. Nat. Acad. Sci.* 109:15; pp. 14069-14074.

van Geert, Paul L. C. (1991). A dynamic systems model of cognitive and language growth. *Psychological Review*,Vol.98. No. 1, pp. 3 - 53.

van Geert, Paul L. C. (2005). Dynamic systems approaches and modeling of developmental processes. In: *Handbook of Developmental Psychology*, Jaan Valsiner and Kevin J. Connolly (eds.) Sage Publications.

van Geert, Paul L. C., and Henderien Steenbeek (2004). A model of behavior modification: The suppression of undesirable behavior. Preprint.

FIGURES

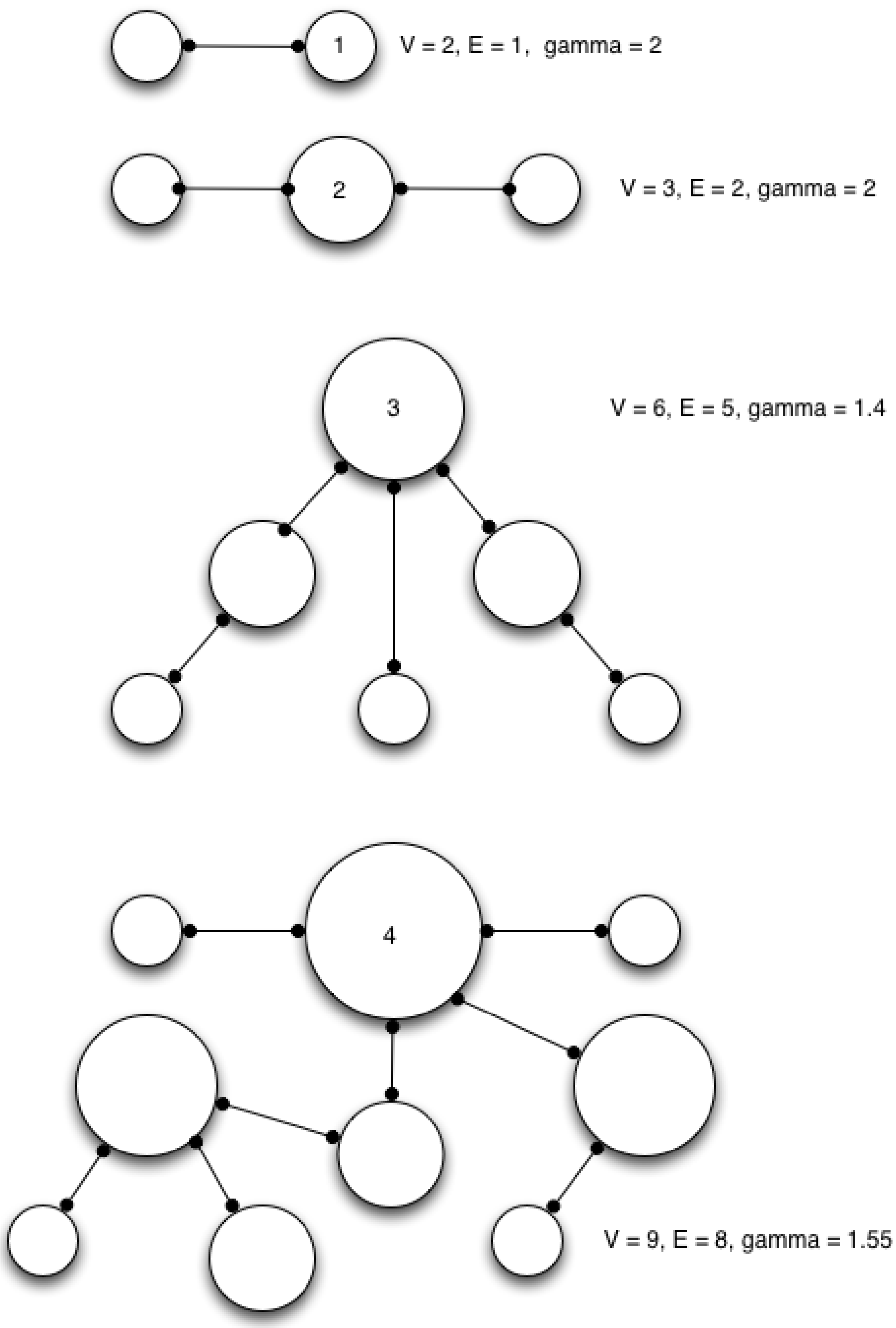

Figure 1: Small power-law graphs

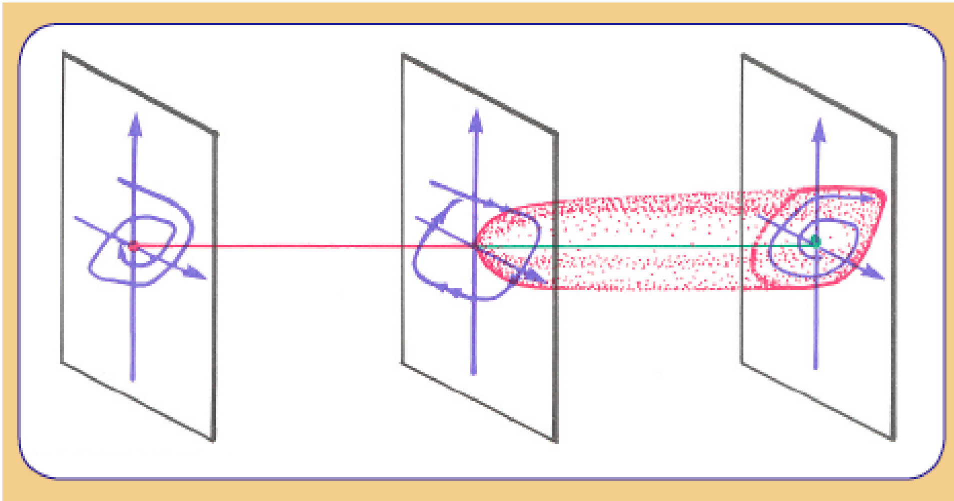

Figure 2: Response diagram of the Hopf bifurcation.

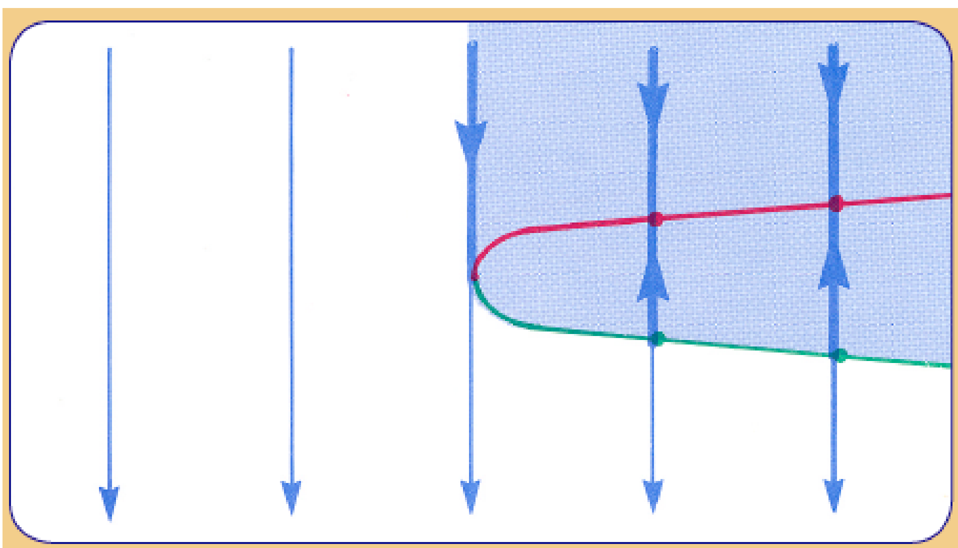

Figure 3: Response diagram of the fold bifurcation.

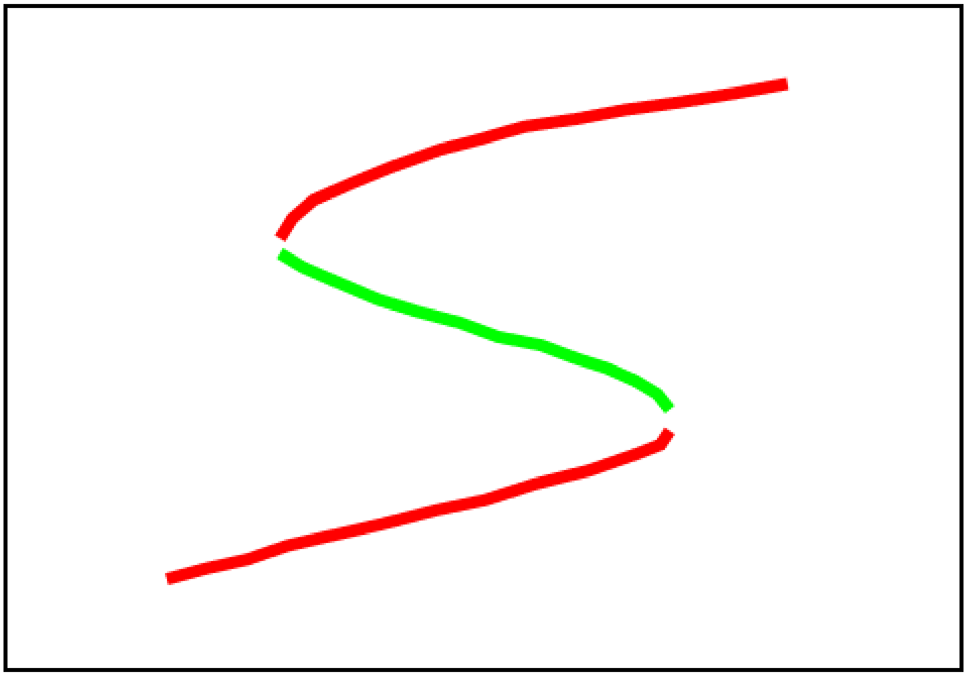

Figure 4: Response diagram of the double-fold scheme.

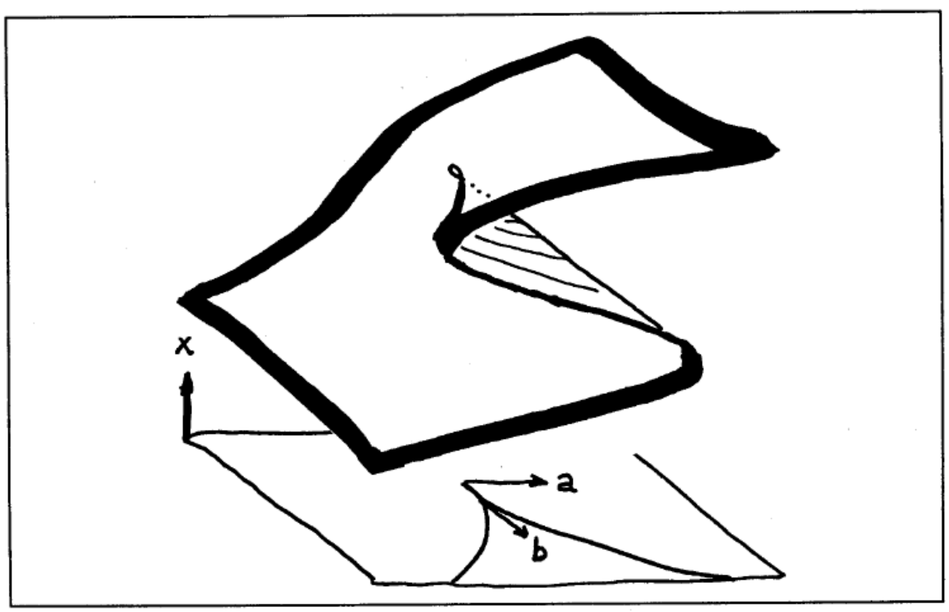

Figure 5: Response diagram of the cusp catastrophe.

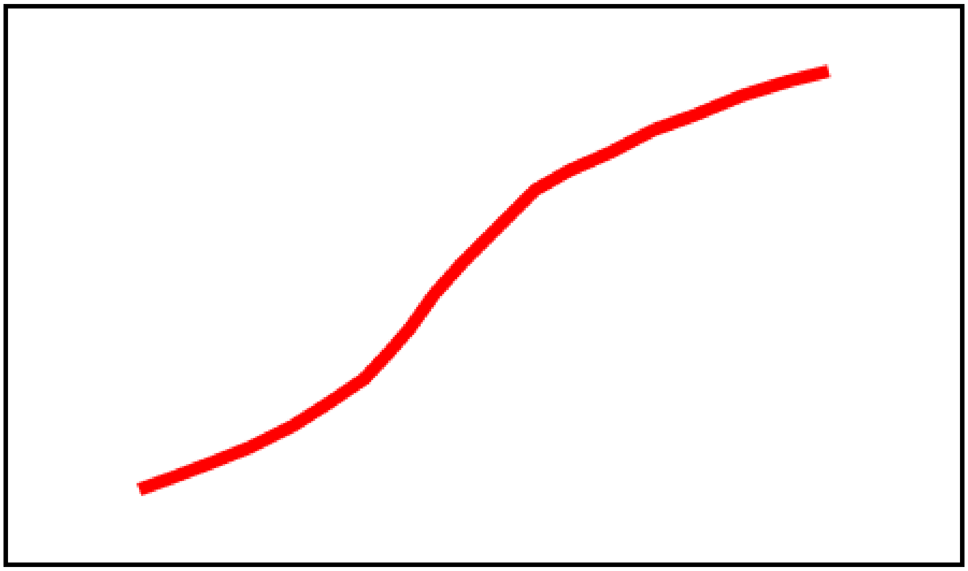

Figure 6: Response diagram with no bifurcation.

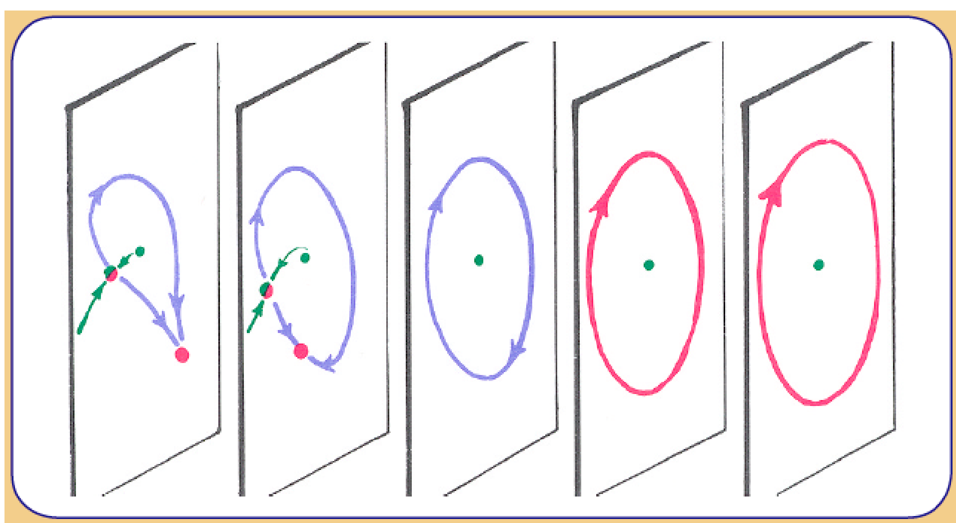

Figure 7: Response diagram of the blue loop bifurcation.

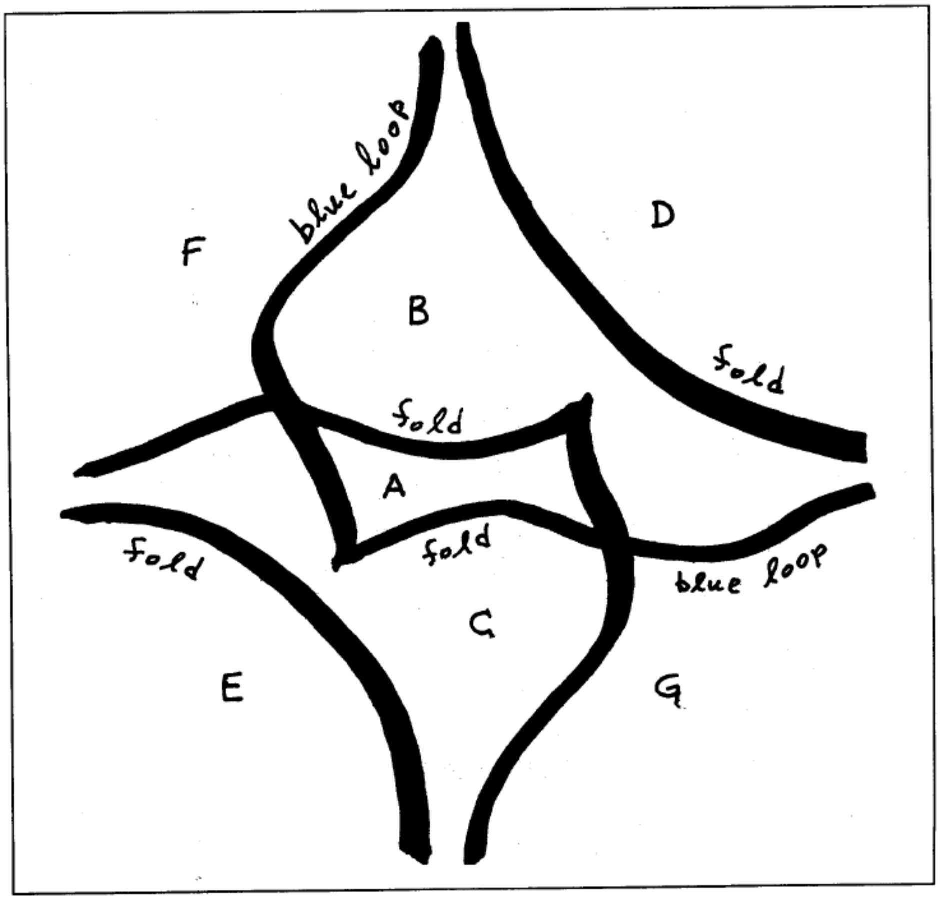

Figure 8: Response diagram of the Kadyrov scheme.

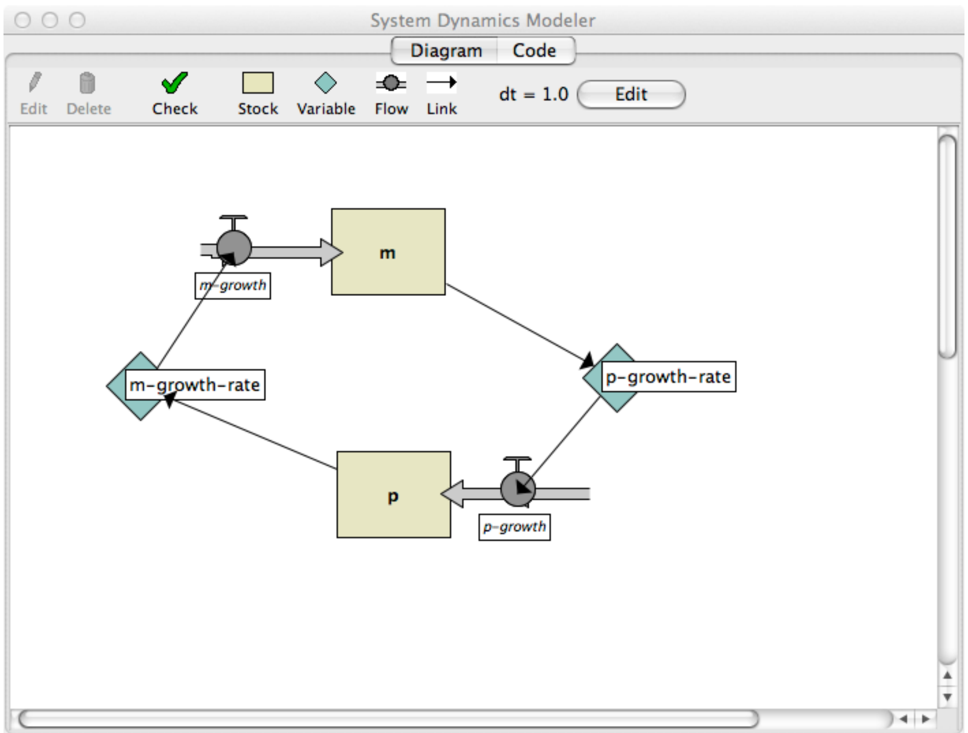

Figure 9: System Dynamics diagram for the BM model.

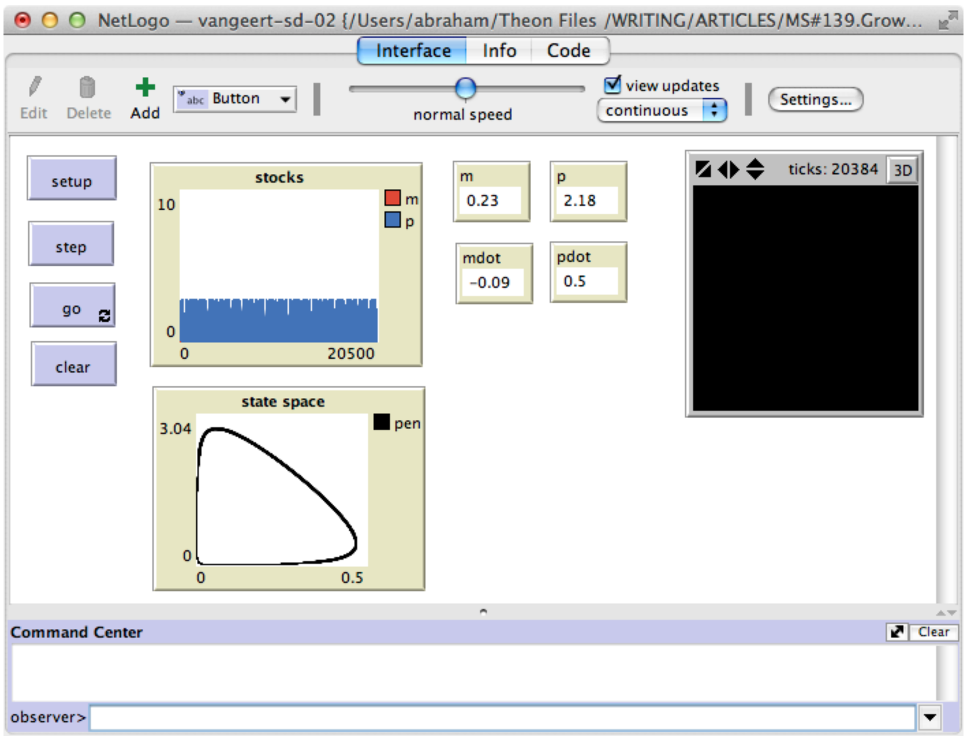

Figure 10: NetLogo User Interface for the BM model.

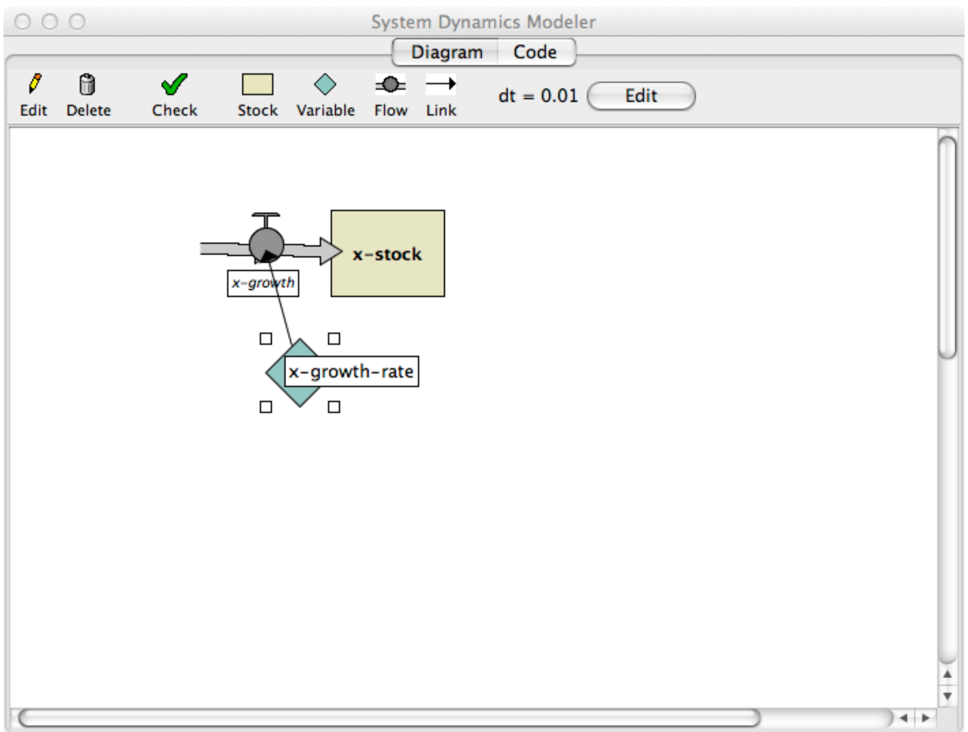

Figure 11: System Dynamics diagram for the one cusp model.

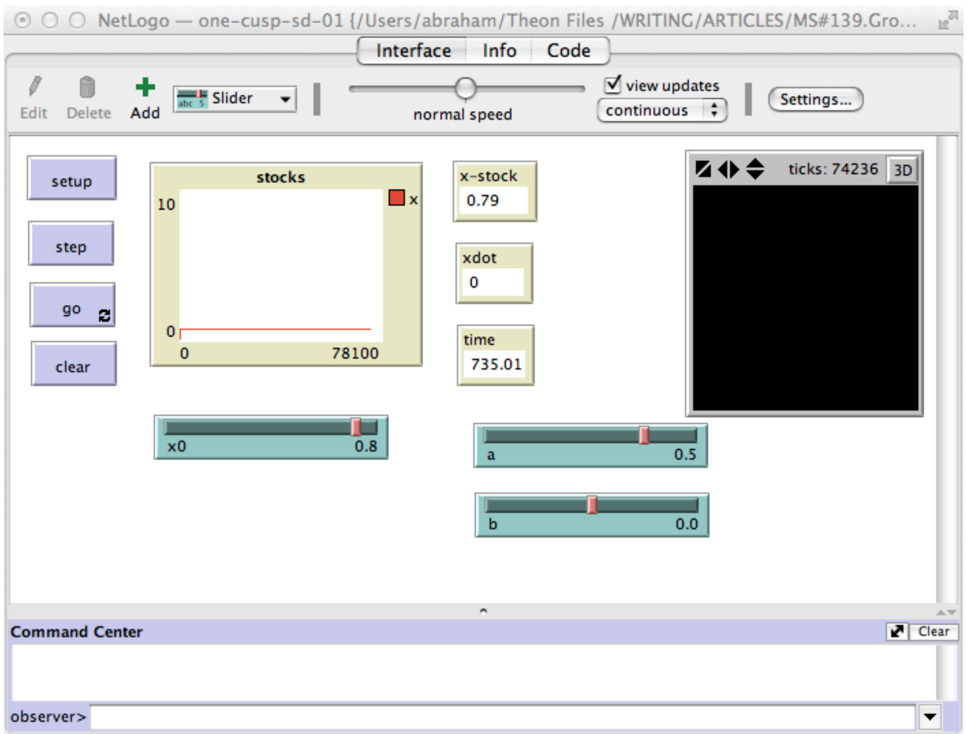

Figure 12: NetLogo User Interface for the one cusp model.

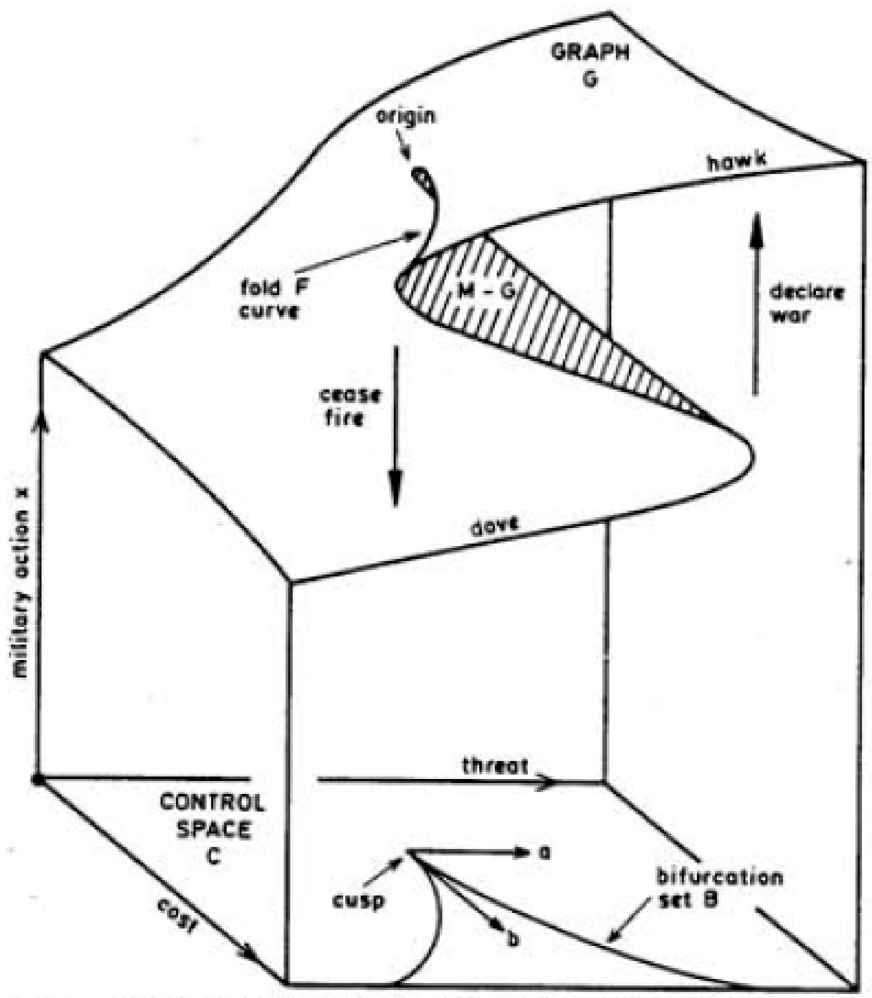

Figure 13: One cusp model for an arms race.

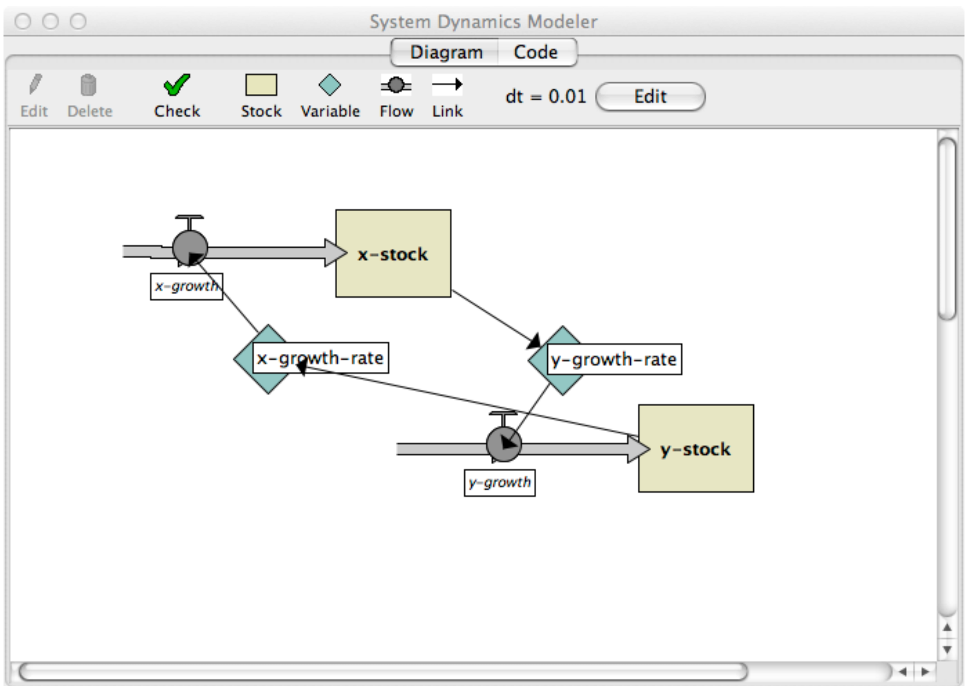

Figure 14: SD diagram for the double cusp model.

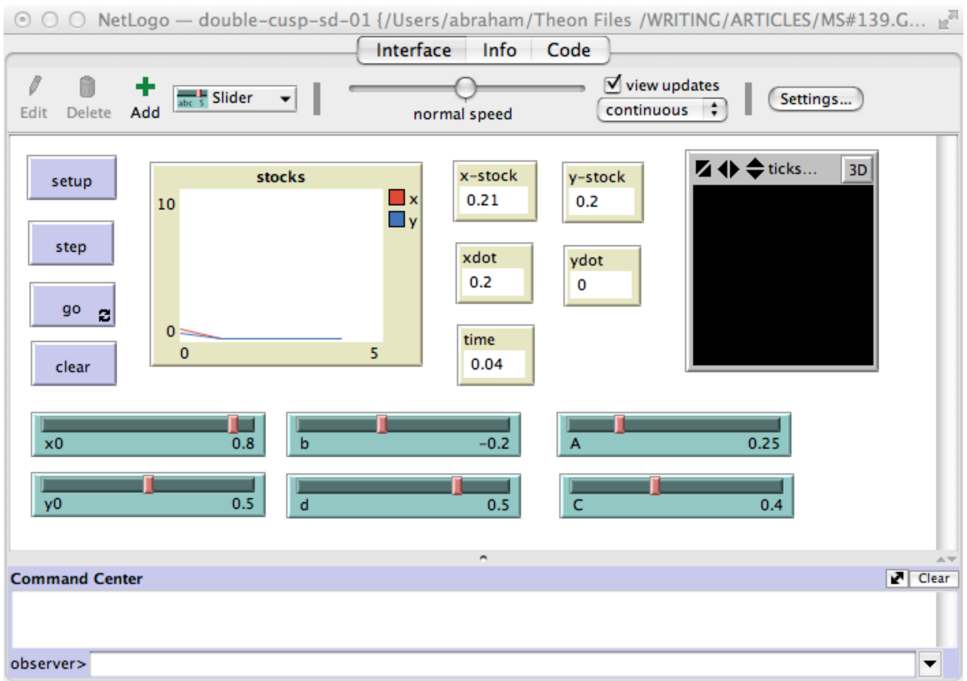

Figure 15: NetLogo User Inerface for the double cusp model.