Published in: *Computers and Mathematics with Applications*

Vol. 26 (1993) pp. 109 — 118

Self-replicating Sequences of Binary Numbers

Wolfgang Banzhaf *

Central Research Laboratory

Mitsubishi Electric Corporation

1-1, Tsukaguchi Honmachi 8-chome, Amagasaki, 661, Japan

ABSTRACT

An algorithm is proposed which allows sequences of binary numbers to interact. We introduce a 2-dimensional matrix form of the sequences achieved by a general folding method. Interactions between 1- and 2-dimensional forms of binary sequences generate new sequences, which compete with the original ones due to selection pressure. Starting from random initial populations, replicating and self-replicating sequences are generated in large numbers. We report on results for 4-digit sequences and propose non-linear differential equations modelling the system.

*Now at: Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139, USA

Self-replication or reproduction has long been considered to be one of the most important prerequisits of life [1]. Since self-replication is a cyclic process it is natural to ask questions about its origin, and how this relates to the origin of life itself [2]. Such questions are only exacerbated by the progress made in the molecular genetic sciences [3]. The simplest living organisms may be certain RNA phages primarily carrying the information necessary to instruct or reprogram host cells to reproduce the phages instead of themselves [4]. However, it is generally accepted that sequences of around 3000 nucleotides are needed for these simplest forms of life to exist.

Some radically simplified artificial systems have been studied in the past in an attempt to elucidate certain aspects of reproduction in well defined mathematical frameworks, such as the cellular automata [5, 6, 7], spinglas models [8, 9] or nonlinear dynamical systems [10, 11, 12, 13]. More recently, complex models have been introduced under the heading of "artificial life" [14, 15, 16].

In this letter we shall introduce another simple system, some members of which show the characteristics of self-replication. It is based on the re-interpretation of logical and mathematical operations and was inspired by findings made nearly a decade ago in biochemistry. Kruger et al. [17] and Guerrier-Takada et al. [18] have demonstrated that strands of RNA, which form 3-dimensional molecules, can operate on themselves and/or other RNA strands. Thus RNA can occasionally assume the double function of being information carrier (represented by the sequence of aminoacids in the strand) *and* enzyme (protein-like, represented by the 2- or

3-dimensional structure of the RNA molecule) at the same time.

In modern digital computers sequences of bits are the primary carrier of information. If we consider this as the genotypic form of an entity we shall call binary strings, we may ask the question whether there exists another form, the phenotype, of a bit string capable of carrying out operations on other bit strings. The answer appears to be 'yes' and the simplest phenotypic form that comes to mind is that of a 2-dimensional matrix.

If we now consider a set of binary symbols $s_i, i = 1, ..., N, N \in \mathcal{N}, s_i \in \{0, 1\}$ [19] to be organized in a 1-dimensional sequence

$$\vec{s} = (s_1, s_2, ..., s_i, ..., s_N)\,, \tag{1}$$

we require the existence of a mapping $\mathcal{M}$

$$\mathcal{M} : \vec{s} \to \mathcal{P}_{\vec{s}} \tag{2}$$

which transforms $\vec{s}$ to a corresponding 2-dimensional matrix form $\mathcal{P}_{\vec{s}}$ of the sequence which should be unique and reversible. The mapping can be termed a folding (this we call the folding method), following the notion used in molecular biology. In this letter we shall restrict ourselves to quadratic matrices of size $\sqrt{N} \times \sqrt{N}$ and corresponding strings of length N. Table 1 gives typical sample results of various mapping (folding) methods applied to strings with N components.

In contrast to another idea for the folding discussed in the literature [20, 21], the phenotypic forms considered here are mathematical operators naturally able to interact with the primary

form of strings.

Assume that an operator $\mathcal{P}_{\vec{s}}$ was formed from string $\vec{s}$. This operator, in turn, can act on another string and generate still another string:

$$\mathcal{P}_{\vec{s}}\ \vec{s}\,' \Rightarrow \vec{s}\,'' \tag{3}$$

We can think of a string as consisting of $\sqrt{N}$ fragments with length $\sqrt{N}$ each. The operator $\mathcal{P}_{\vec{s}}$ is able to transform one of these fragments at a time using semi-local operations. In this way it moves down the string $\vec{s}\,'$ in steps of size $\sqrt{N}$ until it has finally completed the production of a new string $\vec{s}\,''$ (see Fig. 1). Then operator $\mathcal{P}_{\vec{s}}$ unfolds back into its corresponding form as a string $\vec{s}$ and is released, together with $\vec{s}\,'$ and $\vec{s}\,''$, into the ensemble of other strings which will be called *string soup* from now on.

A general operation of $P_{\vec{s}}$ may be subsumed under the following transformation rule:

$$\vec{s}\,'' = \vec{f}\,(\mathcal{P}_{\vec{s}}, \vec{s}\,') \tag{4}$$

A particular example of the action of an operator on a string would be the computation of scalar products. As we deal with binary strings, the following related computation will be studied here:

$$s''_{i+k\sqrt{N}} = \sigma\left[\sum_{j=1}^{j=\sqrt{N}} \mathcal{P}_{\vec{s}\ ij}\ s'_{j+k\sqrt{N}} - \Theta\right] \tag{5}$$
$$i = 1, ..., \sqrt{N} \qquad k = 0, ..., \sqrt{N} - 1$$

where k counts the steps the operator has taken down the string. $\sigma[\ ]$ symbolizes the squashing

function

$$\sigma[x] = \begin{cases} 1 & \text{for } x \geq 0 \\ \\ 0 & \text{for } x < 0 \end{cases} \tag{6}$$

and Θ is an adjustable threshold, here fixed at $\Theta = 1$. To give an example: The $N = 4$ operator $\mathcal{P}_{\vec{s}} = \begin{pmatrix} 0\,0 \\ \\ 0\,1 \end{pmatrix}$ if acting under this operation on string $\vec{s}\,' = (\,1\;0\;0\;1\,)$ would yield a new string $\vec{s}\,'' = (\,0\;0\;0\;1\,)$.

The new interpretation for this operation is to consider it in analogy to a chemical reaction as a polymerization, in which $\mathcal{P}_{\vec{s}}$ reacts with $\vec{s}\,'$ and produces a *new* string $\vec{s}\,''$. In a chemical notation, this would read (skipping the vector arrows from strings)

$$s + s' \longrightarrow s + s' + s'' \tag{7}$$

We therefore require that a new string is assembled using the interaction of s and s' as the source of information that instructs the details of the sequence of s'' [22]. In other words, the proposed system is open with an ongoing generation of new strings. It is this nonequilibrium character of reactions which makes the self-organizing phenomena encountered here possible. Table 2 classifies the various reactions.

There are some options to balance the continued production of new strings which depend on the resource limitations necessarily imposed on such a system. (a) One can run the system with a fixed number of strings. (b) One can do the same after an intitial period of unrestricted growth starting from a small number of strings. (c) One can restrict the raw material in the soup that may be used to build strings. As Eigen [10] has pointed out, all have qualitatively similar effects on the system, namely to force strings into a competition for available resources.

The strings that survive in macroscopic numbers must be able either (i) to reproduce themselves or (ii) to reproduce by the help of others or (iii) to lock into reaction cycles with mutually beneficial transformations. Here, we shall consider systems with a fixed string number M, $M \in \mathcal{N}$.

There are some potentially "lethal" strings in these systems. A string is said to be lethal or "pathological" with respect to the operation of equ. (4) if it is able to replicate in an unproportionally large number in almost any ensemble configuration. In the particular case of equ. (5) the string consisting of "0"'s only is pathological because it is able to replicate with itself and with every other string. We shall call this string *destructor* and shall constantly monitor string soup reactions in order to remove the destructor upon appearance. Another potentially hazardous string consists of "1"'s only. We shall call it the *exploitor*. In addition to being able to replicate itself it is able to replicate with a large fraction of strings. Although the exploitor is pathological we can deal with it in a more gentle way by providing a means of non-deterministic string decay.

To this end, we shall introduce the following general stability criterion for strings: A string may be considered more stable the less "1"'s it contains. Its chance to decay therefore depends on

$$I^{(k)} = \sum_{i=1}^{N} s_i^{(k)}, \qquad k = 1, ..., M. \tag{8}$$

$I^{(k)}$ measures the amount of "1"'s in string k and will determine a probability

$$p^{(k)} = (I^{(k)}/N)^n \tag{9}$$

with which an encountered string should decay. The parameter n shall serve us to adjust prob-

abilities slightly. Note that the exploitor has probability $p = 1$ and must decay upon encounter.

The entire algorithm can now be stated as follows:

STEP 1: Generate M random binary strings of length N each

STEP 2: Select a string[23] and fold it into an operator (a matrix) of dimension $\sqrt{N} \times \sqrt{N}$

STEP 3: Select another string[23] and apply the operator generated in STEP 2

STEP 4: Release the new string, the old string, and the operator (as string) into the string soup

STEP 5: Remove one randomly chosen string in order to compensate for the addition of a string in STEP 4

STEP 6: Monitor the soup and replace destructors by random strings

STEP 7: Select one string[23] and substitute it according to the probability of equation (9) by a random string.

STEP 8: Go to STEP 2

Table 3 shows the impressive amount of possible interactions between strings as we increase their length N. For arbitrary N we have

$$n_S = 2^N - 1 \tag{10}$$

string types and

$$n_R = 2^{2N} - (3 \times 2^N) + 2 \tag{11}$$

reactions, excluding reactions with the destructor and self-reactions. The number of potential self-replications is $n_{SR} = n_S$.

The simplest non-trivial system has strings of length $N = 4$. We shall name them by combining the binary numbers they carry into a decimal number:

$$\vec{s} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} \equiv s^{(k)} \tag{12}$$

with

$$k = \sum_{l=1}^{N} s_l \; 2^{l-1} \tag{13}$$

Everyone of the 15 types of strings can interact with all other types.

We are now in the position to discuss the dynamical behaviour of our system $N = 4$. Global quantities which characterize its time development are the concentrations $x_i(t)$ of all different string sorts $s^{(i)}$:

$$x_i(t) = m_i(t)/M \tag{14}$$

where $m_i(t)$ is the number of actual appearances of string type $s^{(i)}$ in the soup and M, as before, is the constant total number of strings.

Figure 2 shows the first 10^4 iterations through the algorithm with $M = 1000$ strings. Although the dynamics looks very noisy, an impression of attractor state behaviour can be gained.

By increasing the population 100-fold this view is confirmed, as can be seen from Figure 3. The system seems to relax to a macroscopic attractor state as given by the time-averaged concentrations. Running the algorithm under different initial conditions reveals that the macroscopic behaviour is subject to only slight changes, well within the range of fluctuations present in the system.

Using other folding rules or different parameters n, however, results in visible global changes of behaviour [24].

We can try to model the proposed string reactions by a system of coupled differential equations similar to those studied by Eigen and Schuster for the hypercycle [11]. To this end we have to assume that the important aspects of our system can be described by continuous non-random concentration functions $y_i(t)$ of the different string types $i, 1 \leq i \leq n_S$, which are considered to approximate the time averaged concentrations $<x_i>_t$ [25, 26]:

$$y_i(t) \cong <x_i>_t, \qquad 0 \leq y_i(t) \leq 1 \tag{15}$$

The deterministic rate equations in $y_i(t)$ arising through mass action kinetics read:

$$\dot{y}_i(t) = A(t) + \left[B_i y_i(t) + \sum_{k \neq i}^{n_S} C_{ik} y_k(t) - D_i \right] y_i(t) + \sum_{j,k \neq i}^{n_S} W_{ijk} y_j(t) y_k(t) - \frac{y_i(t)}{\sum_k y_k(t)} \Phi(t) \tag{16}$$

where B_i, C_{ik}, W_{ijk} are rate constants for self-replication, replication or other reactions (see Table 2) derived from a reaction table of the participating string types. They are either 1 if the reaction exists or 0 otherwise. D_i determines a selection term

$$D_i = p^{(i)}. \tag{17}$$

$A(t)$ is an unspecific growth term

$$A(t) = \sum_{i,j}^{n_S} a_{ij} y_i(t) y_j(t) + \sum_{i}^{n_S} D_i y_i(t) \tag{18}$$

where

$$a_{ij} = \begin{cases} 1 & \text{if the reaction between } s^{(i)} \text{ and } s^{(j)} \text{ produces } s^{(0)} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

reflects reactions producing the destructor. Finally, $\Phi(t)$ is a flow term used to enact competition between the various string sorts $s^{(i)}$ by enforcing that the overall sum of concentrations in the system is kept constant. The behaviour of these equations for 15 string types with concentrations $y_i(t)$ and rate constants derived from the $N = 4$ system is shown in Figure 4. We can clearly observe that some concentrations merge into the same levels, due to the particular interactions present in the dynamics of this binary string system. Since the number of string types is very low in this simulation, no type dies out. The comparison between the statistical data and the numerical integration of (16) shows very good agreement.

A wealth of models becomes possible in the proposed framework. Here we have only discussed the simplest system in some detail. More complex systems will be treated elsewhere [24, 27]. In general, the combinatorial explosion we encounter here will be useful for application purposes like optimization [28], since we are able to act at the "atomic level" of the system (the binary numbers). In addition, the folding methods introduced here can be refined as to allow the local structure of a sequence to partially determine its 2-dimensional form.

Acknowledgement

I wish to thank my colleagues for interesting discussions. Special thanks go to T. Iwamoto, who reignited my interest in autocatalytic systems. I am grateful to Drs. T. Nakayama and K. Kyuma for establishing and maintaining the creative atmosphere in our research group. Finally I thank Dr. J. Bell for carefully checking the style of my manuscript.

References

[1] Schrödinger, E., *What is Life?*, Cambridge University Press, UK, 1944

[2] Miller, S. and Orgel, L.E., *Origins of Life on Earth*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1974

[3] Watson, J.D., *Molecular Biology of the Gene*, 3rd ed. W.A. Benjamin, Menlo Park, CA, USA, 1976

[4] Zinder N.D (Ed.), *RNA-Phages*, Cold Spring Harbor Monograph Series, Cold Spring Harbor Laboratories, USA, 1975

[5] von Neumann, J., *Theory of Self-Reproducing Automata*, ed. A. W. Burks, University of Illinois, USA, 1966

[6] Berlekamp, E.R., Conway, J.H. and Guy, R.K., *Winning Ways for your mathematical plays*, Vol 2., Academic Press, NY, USA, 1984

[7] Wolfram, S., *Theory and Applications of Cellular Automata*, World Scientific, Singapore, 1986

[8] Anderson, P.W., Proc. Natl. Acad. Sci. USA, **80** (1983) 3386

[9] Abbott, L.F., J. Mol. Evol. **27** (1988) 114

[10] Eigen, M., Naturwissenschaften **58** (1971) 465

[11] Eigen, M. and Schuster, P., Naturwissenschaften **64** (1977) 541, **65** (1978) 7, **65** (1978) 341

[12] Kauffman, S., J. Theor. Biol. **119** (1987) 1

[13] Schuster, P., Phys. Scr. **35** (1987) 402

[14] Langton, C.G., (ed.), *Artificial Life*, Santa Fe Institute Studies on the Sciences of Complexity, Proc. Vol. VI, Addison-Wesley, Reading, MA, 1989

[15] Ray, T.S., in: *Artificial Life II*, Langton, C.G., Taylor, C., Farmer, J.D., Rasmussen, S. (eds.), Santa Fe Institute Studies on the Sciences of Complexity, Proc. Vol. X, Addison-Wesley, Reading, MA, 1991

[16] Fontana, W., in: *Artificial Life II*, Langton, C.G., Taylor, C., Farmer, J.D., Rasmussen, S. (eds.), Santa Fe Institute Studies on the Sciences of Complexity, Proc. Vol. X, Addison-Wesley, Reading, MA, 1991

[17] Kruger, K., Grabowski, P., Zaug, A., Sands, J., Gottschling, D. and Cech, T., Cell **31** (1982) 147

[18] Guerrier-Takada C., Gardiner, K., Marsh, T., Pace, N. and Altman, S., Cell **35** (1983) 849

[19] We are not restricted to binary numbers. Any numbers and possibly sets of symbols could be used provided an appropriate and natural interaction mechanism can be found.

[20] Fontana, W. and Schuster, P., Biophys. Chem. **26** (1987) 123

[21] Fontana, W., Schnabl, W. and Schuster, P., Phys. Rev. **A40** (1989) 3301

[22] It is also possible that only one of the two, either string or operator, should be conserved. Qualitatively the behaviour of the system is similar.

[23] Selection is at random, from a uniform distribution.

[24] Banzhaf, W., Technical Report, Central Research Laboratory, Mitsubishi Electric Corporation (to be published)

[25] Gillespie, D.T. and Mangel, M., J. Chem. Phys. **75** (1981) 704

[26] Kurtz, T.G., J. Chem. Phys. **57** (1972) 2976

[27] Banzhaf, W., *Self-replicating binary strings - The build-up of complexity*, Complex Systems (submitted)

[28] Banzhaf, W., *Self-replicating binary strings - Applications*, (to be published)

Figure Captions

Figure 1: An operator $\mathcal{P}_{\vec{s}'}$ acts upon a string $\vec{s}'$ to produce a new string $\vec{s}''$.

Figure 2: Type concentrations during the first $10,000$ iterations through the algorithm of a binary string system, $N = 4, M = 1000$. The initial configuration is a randomly distributed population. Noisy attractor system.

Figure 3: Type concentrations in a simulation using the reaction matrix of the binary string system $N = 4$. $M = 100,000$ strings were simulated.

Figure 4: Simulation of the differential equations (16). Constants derived from binary string system $N = 4$. Initial conditions same as in Figure 3. Types 1,2,4,8 show highest, types 7,11,13,14 show lowest concentration.

Table 1: Results of 4 different folding methods applied to a string $\vec{s}$. Length N is a square number.

Table 2: Classification of different polymerization reactions. Raw material X not mentioned. Reactants are conserved.

Table 3: Some low dimensional examples. $\sqrt{N}$: Matrix size in one dimension; N: Length of strings; n_S: Number of different string types, excluding destructor; n_R: Number of possible reactions, excluding self-reactions.

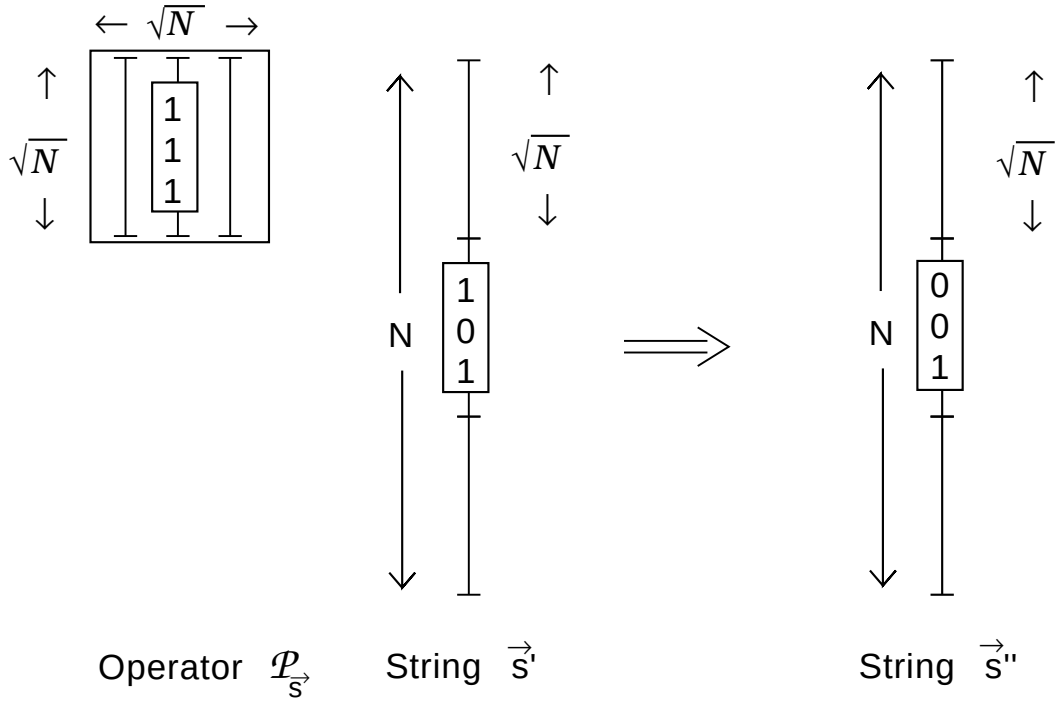
$\leftarrow \sqrt{N} \rightarrow$
$\uparrow$
$\sqrt{N}$
$\downarrow$
1
1
1
N
1
0
1
$\uparrow$
$\sqrt{N}$
$\downarrow$
$\Longrightarrow$
N
0
0
1
$\uparrow$
$\sqrt{N}$
$\downarrow$
Operator $\mathcal{P}_{\vec{s}}$
String $\vec{s}$'
String $\vec{s}$''

Figure 1:

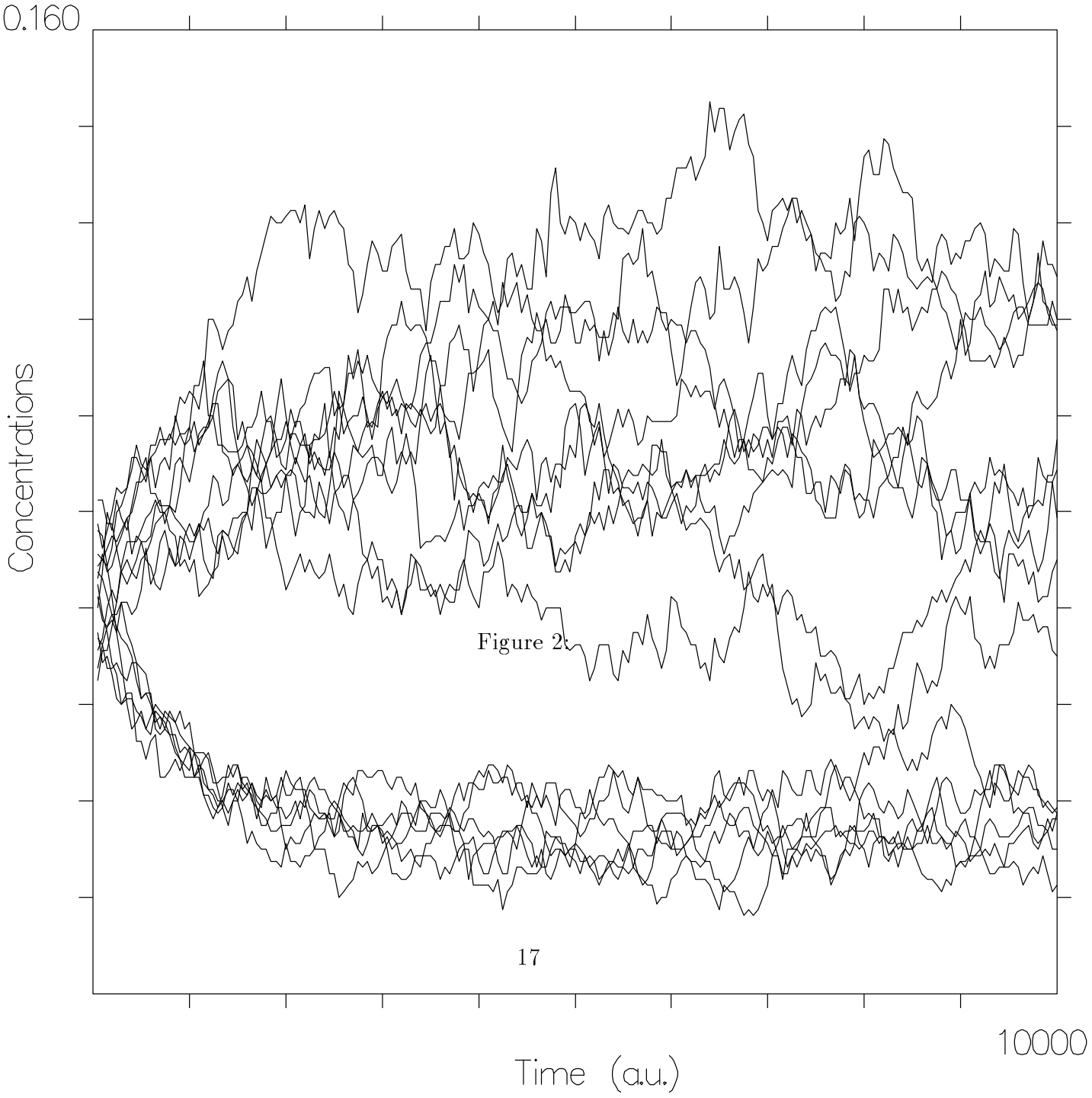

Figure 2:

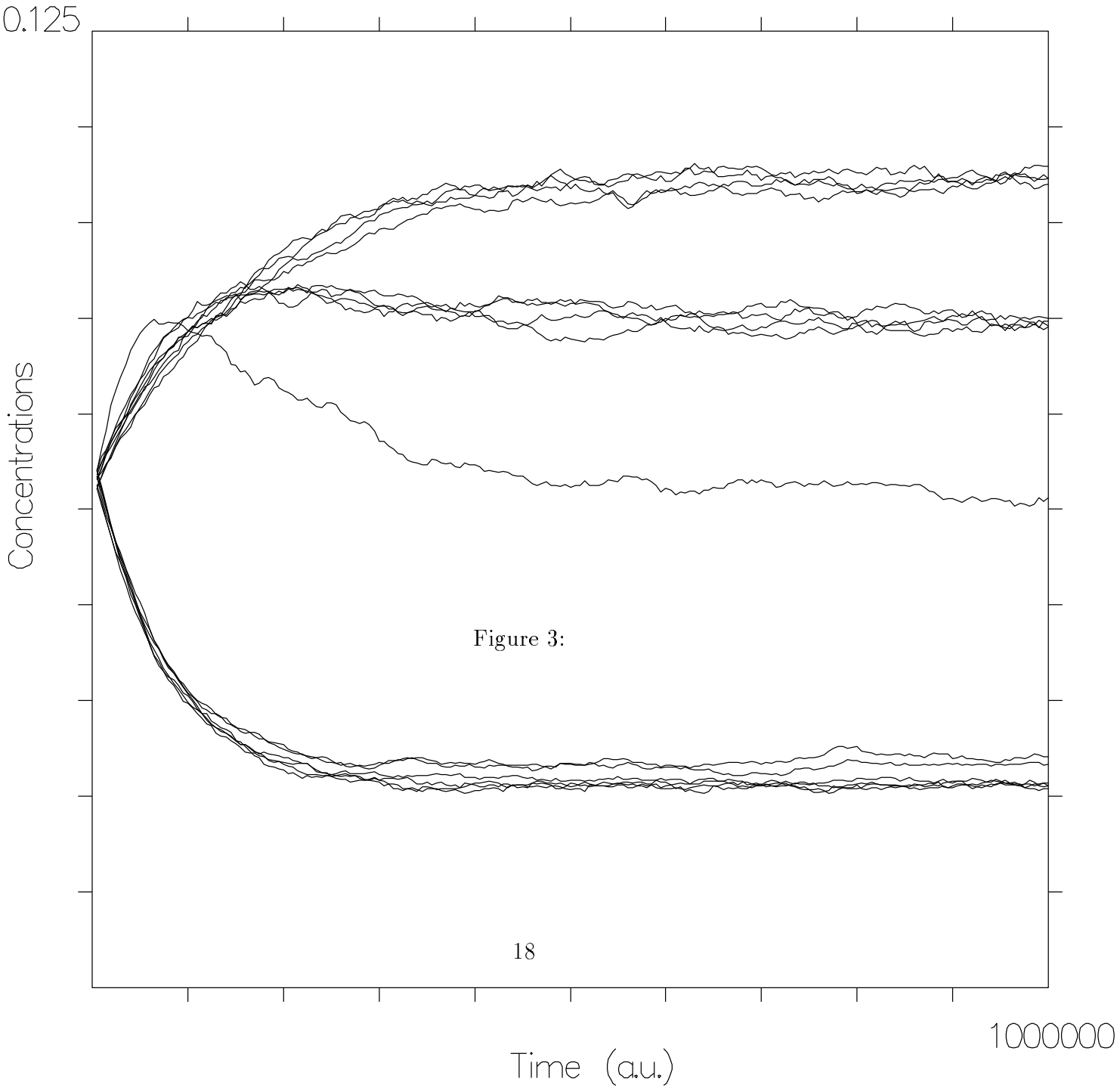

Figure 3:

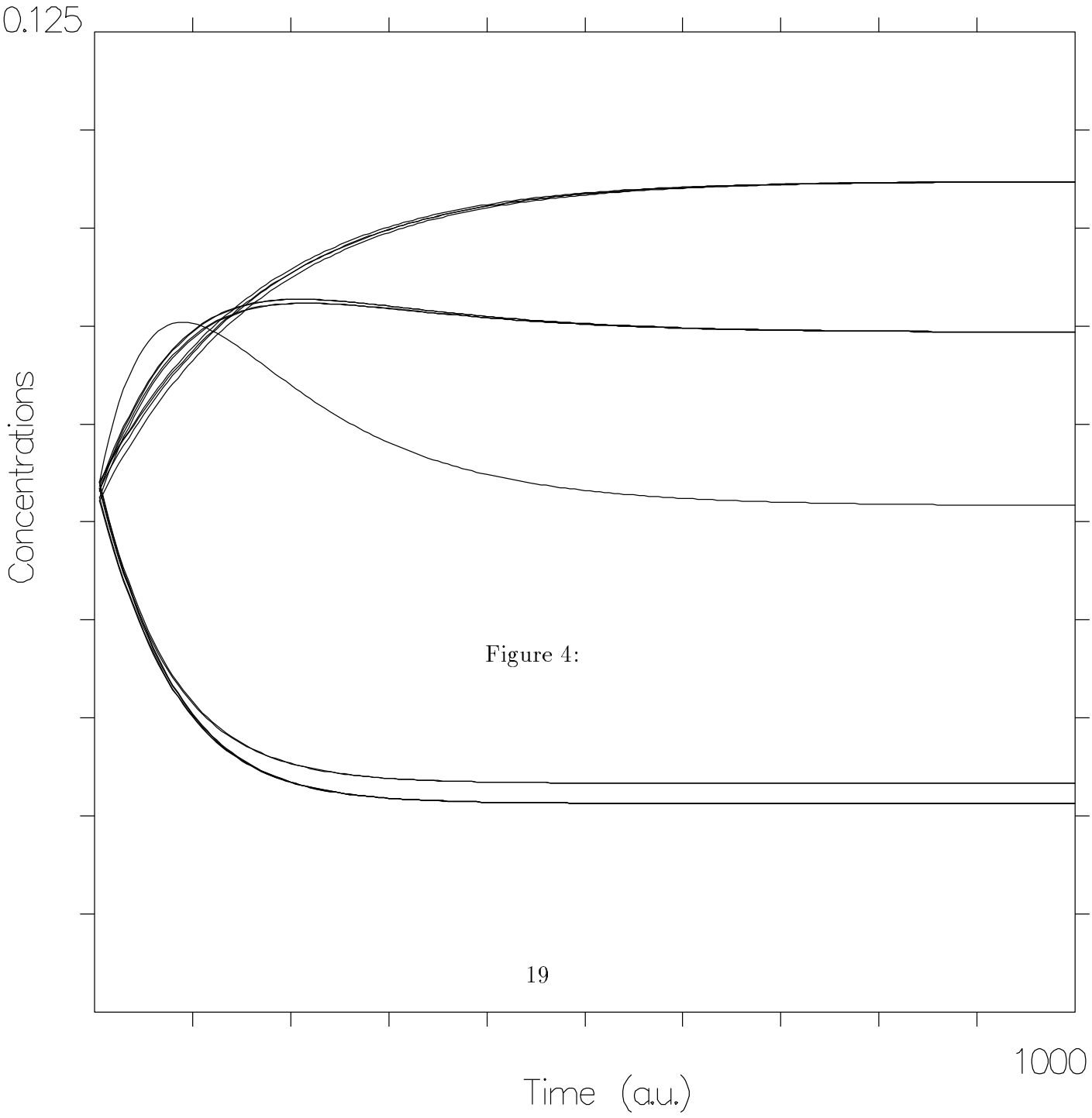

Figure 4:

Method	1	2
Outcome	$\begin{pmatrix} s_1 & \cdots & s_{\sqrt{N}} \\ s_{\sqrt{N}+1} & \cdots & s_{2\cdot\sqrt{N}} \\ \vdots & \ddots & \vdots \\ s_{(\sqrt{N}-1)\cdot\sqrt{N}+1} & \cdots & s_N \end{pmatrix}$	$\begin{pmatrix} s_1 & s_{\sqrt{N}+1} & \cdots & s_{(\sqrt{N}-1)\cdot\sqrt{N}+1} \\ \vdots & \vdots & \ddots & \cdots \\ s_{\sqrt{N}} & s_{2\cdot\sqrt{N}} & \cdots & s_N \end{pmatrix}$
Method	3	4
Outcome	$\begin{pmatrix} s_1 & \cdots & s_{\sqrt{N}} \\ s_{2\cdot\sqrt{N}} & \cdots & s_{\sqrt{N}+1} \\ s_{2\cdot\sqrt{N}+1} & \cdots & s_{3\cdot\sqrt{N}} \\ \vdots & \vdots & \vdots \end{pmatrix}$	$\begin{pmatrix} s_1 & s_{2\cdot\sqrt{N}} & s_{2\cdot\sqrt{N}+1} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ s_{\sqrt{N}} & s_{\sqrt{N}+1} & s_{3\cdot\sqrt{N}} & \cdots \end{pmatrix}$

Table 1:

Reactants	Product	Description
$s + s'$	s''	other reaction
$s + s'$	s	replication
$s + s'$	s'	replication
$s + s$	s'	other reaction
$s + s$	s	self-replication

Table 2:

$\sqrt{N}$	2	3	4	5	10
N	4	9	16	25	100
n_S	15	511	65535	$\sim 10^7$	$\sim 10^{30}$
n_R	210	$\sim 2.6 \cdot 10^5$	$\sim 4 \cdot 10^9$	$\sim 10^{15}$	$\sim 10^{60}$

Table 3: