



ELSEVIER

Simulation Practice and Theory 7 (1999) 309–324

**SIMULATION  
PRACTICE AND THEORY**

www.elsevier.nl/locate/simpra

# Simulations of asynchronous evolution of discrete systems

Jacques M. Bahi<sup>a</sup>, Christian J. Michel<sup>b,\*</sup>

<sup>a</sup> *Laboratoire de Calcul Scientifique, Antenne de Belfort, UMR CNRS 6623, BP 527, 90016 Belfort, France*

<sup>b</sup> *Laboratoire de Recherche en Informatique, Institut Polytechnique de Sévenans, Rue du Château à Sévenans, 90010 Belfort, France*

Received 3 June 1998; received in revised form 1 February 1999

---

## Abstract

Recently, a convergence theorem of asynchronous iterations of discrete dynamic systems partitioned into blocks has been proved [2]. This theorem is verified with several asynchronous block strategies. It also generalizes the chaotic iterations. Different simulations of asynchronous evolution of discrete systems performed with the research software Discrete System Evolution (DSE), lead to the first experimental results predicted by this theorem. © 1999 Elsevier Science B.V. All rights reserved.

*Keywords:* Boolean iterations; Asynchronous algorithms; Discrete dynamic systems; Numerical simulations; Research software

---

## 1. Introduction

The convergence results in the asynchronous continuous framework are well-known and are based on a contraction hypothesis with respect to a maximum norm, see e.g. [4,6,5,8,1,3]. However, this approach cannot be applied to the discrete framework. Indeed, this hypothesis in the continuous framework leads to constant functions in the discrete framework and a new study is necessary for the discrete case. We have supposed that  $F$  is a contraction with respect to a vectorial distance [2]. The contraction with respect to a vectorial distance was first introduced by [7] in order to study a particular case of asynchronous iterations, namely the chaotic ones.

The discrete system considered here has  $n$  components. Each component  $i$  takes a finite number of values  $x_i, i \in \{1, \dots, n\}$ . This system is partitioned into  $\alpha$  blocks.

---

\* Corresponding author. Fax: 33 3 84 58 30 30; e-mail: christian.michel@utbm.fr

Each block  $i$  has  $n_i$  components with  $\sum_{i=1}^z n_i = n$ . The value of a block  $i$  is denoted by  $X_i$  and the value of the block system, by  $X = (X_1, \dots, X_z)$ . The dynamic of the system is described according to a function  $f$

$$f(x) = (f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n)).$$

This function  $f$  is partitioned into a compatible way, i.e.

$$f(x) = F(X) = (F_1(X_1, \dots, X_z), \dots, F_z(X_1, \dots, X_z)).$$

The state of a system (resp. a block system) at the time  $t$  is represented by  $x^t$  (resp.  $X^t$ ), or, more precisely by

$$x^t = X^t = (x_1^t, \dots, x_n^t) = (X_1^t, \dots, X_z^t).$$

**Definition 1.1.** Consider the Cartesian product  $E = \prod_{i=1}^n E_i$ , where  $E_i$  represents the finite set of possible values  $x_i$  of the component  $i$  of the discrete system. The vectorial distance  $d: E \times E \rightarrow \{0, 1\}^n$  is defined by for all  $(x, y) \in E \times E$

$$(x, y) \rightarrow d(x, y) = (\delta(x_1, y_1), \dots, \delta(x_n, y_n)),$$

where

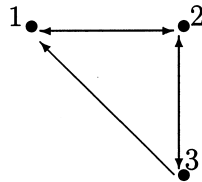
$$\delta(x_i, y_i) = \begin{cases} 1 & \text{if } x_i \neq y_i, \\ 0 & \text{if } x_i = y_i. \end{cases}$$

**Definition 1.2.** Consider a discrete system whose dynamic is described according to a function  $f: E \rightarrow E$ . The boolean matrix  $B(f)$  associated with  $f$  is defined by its general term  $b_{ij}, i, j \in \{1, \dots, n\}$ , so that

$$b_{ij} = \begin{cases} 1 & \text{if the } i\text{th component of } f \text{ depends on } x_j, \\ 0 & \text{otherwise.} \end{cases}$$

As  $B(f)$  is a boolean matrix, its only possible eigenvalues are 0 or 1.

**Example 1.1.** The following simple example shows that to have the boolean matrix  $B(f)$  is equivalent to have the connexion graph of the discrete system. Consider a discrete system with three components 1, 2 and 3 (cellular automata, processors, neurons, etc). Assume that the notation  $1 \rightarrow 2$  means that component 1 informs component 2. If these components are connected as shown in the figure below



then, the matrix  $B(f)$  associated with any mapping  $f = (f_1, f_2, f_3)$  describing the dynamic of the system according to the above graph, is

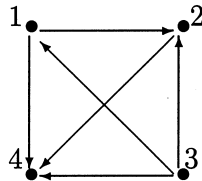
$$B(f) = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

**Proposition 1.1** (detailed in [7]).  $d(f(x), f(y)) \leq B(f)d(x, y)$ , for all  $(x, y) \in E \times E$ , where the componentwise order relation  $\leq$  is defined in  $\{0, 1\}^n$  by  $0 \leq 0 \leq 1 \leq 1$ .

**Definition 1.3.** The function  $f$  is a contraction if the spectral radius  $\rho(B)$  of the associated matrix  $B(f)$  is equal to 0.  $B(f)$  is called the contraction matrix of  $f$ .

**Proposition 1.2** (detailed in [7]). If the function  $f$  is a contraction on  $E = \prod_{i=1}^n E_i$  then there exists a unique  $x^* \in E$  so that  $x^* = f(x^*)$ .  $x^*$  is called the fixed point of  $f$ .

**Remark 1.1.** The contraction concept is related to the connexion graph of the discrete system. Indeed, an iteration function  $f$  describing the dynamic of the system is a contraction if and only if its connexion graph has no cycles. Note that there are an infinity of iteration functions associated with a given connexion graph.



All iteration functions associated with this connexion graph are contractions.

**Definition 1.4.** The block vectorial distance is defined as follows

$$d(X, Y) = (\delta(X_1, Y_1), \dots, \delta(X_\alpha, Y_\alpha))$$

such that

$$\delta(X_i, Y_i) = \begin{cases} 1 & \text{if } X_i \neq Y_i, \\ 0 & \text{if } X_i = Y_i. \end{cases}$$

**2. Totally asynchronous discrete model**

**Definition 2.1.** Let the strategy  $\{J(t)\}_{t \in \mathbb{N}}$  be a sequence of non-empty subsets of  $\{1, \dots, \alpha\}$  at the time  $t$ . Let  $\{s_j^i(t)\}_{t \in \mathbb{N}}, i, j \in \{1, \dots, \alpha\}$ , be a sequence of integers at the time  $t$  satisfying the three following conditions:

- (i)  $s_j^i(t) = t - r_j^i(t)$  with  $0 \leq r_j^i(t) \leq t, r_j^i(t)$  being the delay of the block  $j$  compared to the block  $i$ . These delays may be generated by different communications and computation sizes of the elements of the system. This model takes no synchronization hypothesis.
- (ii)  $\forall i, j \in \{1, \dots, \alpha\}, \lim_{t \rightarrow \infty} s_j^i(t) = \infty$ , i.e. the delays associated with the block  $i$  are unbounded but follow the iterations of the system.

(iii)  $\forall i \in \{1, \dots, \alpha\}, \text{Card}(\{t, i \in J(t)\}) = \infty$ , i.e. no block is definitively lost.

Then, the asynchronous iterations with delays  $\{r_j^i(t)\}$  according to the strategy  $\{J(t)\}$  are described by the algorithm

$$\begin{aligned} &\text{Given } X^0 = (X_1^0, \dots, X_\alpha^0) \\ &t = 0, 1, \dots \\ &\quad i = 1, \dots, \alpha \\ &X_i^{t+1} = \begin{cases} F_i(X_1^{s_1^i(t)}, \dots, X_\alpha^{s_\alpha^i(t)}) & \text{if } i \in J(t) \\ X_i^t & \text{if } i \notin J(t). \end{cases} \end{aligned} \quad (1)$$

**Remark 2.1.**

1. The block  $i$  at the time  $t$  is either updated (iterated) by using some blocks  $j$  with states  $X_j^{s_j^i(t)}$  available at the previous time  $s_j^i(t) = t - r_j^i(t) \leq t$ , i.e.  $X_i^{t+1} = F_i(X_1^{s_1^i(t)}, \dots, X_\alpha^{s_\alpha^i(t)})$ , or not updated, i.e.  $X_i^{t+1} = X_i^t$ .
2. These two alternatives are described by the iteration strategy  $\{J(t)\}_{t \in \mathbb{N}}$ ,  $J(t) \subset \{1, \dots, \alpha\} \forall t \in \mathbb{N}$ : if the block  $i$  belongs to the strategy  $J(t)$  then its state  $X_i^{t+1}$  is updated by  $F_i$  otherwise the block  $i$  is not updated. The iterations considered are asynchronous, i.e. chaotic iterations with delays.
3. The chaotic iterations are particular cases of Alg. (1) with  $s_j^i(t) = t$ .

**Remark 2.2.** The set of all possible strategies is not countable. Furthermore, if the delays are also considered, the evolution complexity of asynchronous iterations is hard to imagine.

**Example 2.1.** This simple example gives an evolution of a discrete model at three components with chaotic and asynchronous iterations. It will show in particular that

- (i) a successive substitution function  $F$  with a fixed point does not automatically imply the convergence of chaotic and asynchronous iterations;
- (ii) the chaotic iterations are particular cases of asynchronous iterations;
- (iii) the asynchronous iterations do not represent a classical process of successive substitutions;
- (iv) asynchronous iterations with different strategies and delays can reach a state which is the fixed point in the successive substitutions, but without convergence.

Let the function  $F$  given by the following Table 1.

Then, the graph of successive substitutions is given in Fig. 1.

The graph of  $F$  is very simple with a fixed point 3 (no cycles) (Fig. 1). For simplicity reasons, the chaotic and asynchronous iterations considered below are componentwise iterations, i.e. at each iteration, only one component is updated. The chaotic function  $F_i(X)$  is given by the following Table 2.

Then, the graph of chaotic iterations is given in Fig. 2.

Although there is a fixed point in the successive substitutions, the chaotic iterations do not converge. For example, if the initial state is  $4 = (1, 0, 0)$ , then the chaotic iterations are in a cycle containing the states 4, 6, 7 (Fig. 2).

Table 1  
Example of successive substitutions

State	$X$	$F(X)$
0	0,0,0	0,0,1
1	0,0,1	1,0,1
2	0,1,0	0,0,1
3	0,1,1	0,1,1
4	1,0,0	1,1,0
5	1,0,1	0,1,1
6	1,1,0	1,0,1
7	1,1,1	1,1,0

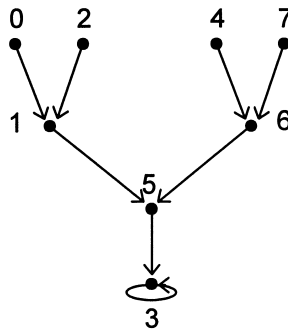


Fig. 1. Graph of successive substitutions associated with Table 1.

Let  $X^s$  be a previous value of  $X$ .  $F_i(X^s)$  is a value among all possible values of  $F_i(X)$ , i.e. 0 or 1 in this example. The asynchronous function  $F_i(X^s)$  is given by the following Table 3.

Then, the graph of asynchronous iterations is given in Fig. 3.

The chaotic iterations are obviously particular cases of asynchronous iterations (Figs. 2 and 3). Therefore, the asynchronous iterations do also not converge. Other-

Table 2  
Example of chaotic iterations

State	$X$	$F(X)$	$F_1(X), X_2, X_3$	$X_1, F_2(X), X_3$	$X_1, X_2, F_3(X)$
0	0,0,0	0,0,1	0,0,0	0,0,0	0,0,1
1	0,0,1	1,0,1	1,0,1	0,0,1	0,0,1
2	0,1,0	0,0,1	0,1,0	0,0,0	0,1,1
3	0,1,1	0,1,1	0,1,1	0,1,1	0,1,1
4	1,0,0	1,1,0	1,0,0	1,1,0	1,0,0
5	1,0,1	0,1,1	0,0,1	1,1,1	1,0,1
6	1,1,0	1,0,1	1,1,0	1,0,0	1,1,1
7	1,1,1	1,1,0	1,1,1	1,1,1	1,1,0

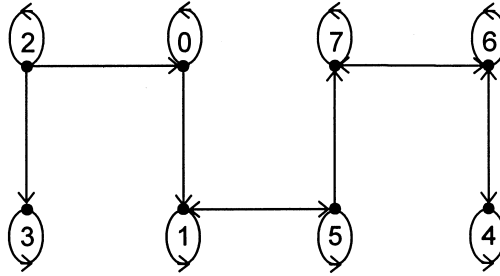


Fig. 2. Graph of chaotic iterations associated with Table 2.

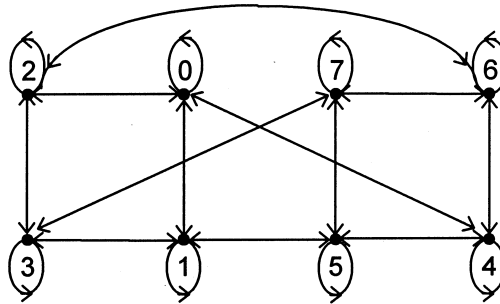


Fig. 3. Graph of asynchronous iterations associated with Table 3.

wise, asynchronous iterations with different strategies and delays can reach state 3, the fixed point in the successive substitutions, but without convergence (Fig. 3).

This example of evolution also explains that the asynchronous iterations do not constitute a process of successive substitutions. Table 4 below shows a particular asynchronous evolution of the model corresponding to a particular choice of a strategy  $J(t)$  and delays  $s_j^i(t)$ .

**Definition 2.2.** Consider the strictly increasing sequence of integers  $\{p_l\}_{l \in \mathbb{N}}$  as follows

$$\begin{aligned}
 &p_0 = 0 \\
 &p_{l+1} \text{ is the smallest integer satisfying} \\
 &\quad \bigcup_{p_l \leq s_{\min}(t) \leq t < p_{l+1}} J(t) = \{1, \dots, \alpha\} \\
 &\quad \text{where } J(t) \subset \{1, \dots, \alpha\} \text{ and } s_{\min}(t) = \min_{1 \leq i, j \leq \alpha} \{s_j^i(t)\}.
 \end{aligned}$$

This sequence  $\{p_l\}$  is well defined thanks to the conditions (ii) and (iii) of Definition 2.1.

**Theorem 2.1** (see [2]). *Let a discrete dynamic system of  $n$  components be partitioned into  $\alpha$  blocks and described by an iteration function  $F = (F_1, \dots, F_\alpha)$ . If  $F$  is a contraction with respect to the block vectorial distance of Definition 1.4 on the finite*

Table 3  
Example of asynchronous iterations

State	$X$	$F(X)$	$F_1(X), X_2, X_3$	$X_1, X_2, F_3(X)$	$X_1, F_2(X), X_3$	$X_1, X_2, F_3(X)$	$F_1(X^s), X_2, X_3$	$X_1, F_2(X^s), X_3$	$X_1, X_2, F_3(X^s)$
0	0,0,0	0,0,1	0,0,0	0,0,1	0,0,0	0,0,0	1,0,0	0,1,0	0,0,0
1	0,0,1	1,0,1	1,0,1	0,0,1	0,0,1	0,0,0	0,0,1	0,1,1	0,0,0
2	0,1,0	0,0,1	0,1,0	0,0,0	0,0,0	0,1,0	1,1,0	0,1,0	0,1,0
3	0,1,1	0,1,1	0,1,1	0,1,1	0,1,1	0,1,0	1,1,1	0,0,1	0,1,0
4	1,0,0	1,1,0	1,0,0	1,1,0	1,1,0	1,0,0	0,0,0	1,0,0	1,0,1
5	1,0,1	0,1,1	0,0,1	1,1,1	1,1,1	1,0,1	1,0,1	1,0,1	1,0,0
6	1,1,0	1,0,1	1,1,0	1,0,0	1,0,0	1,1,1	0,1,0	1,1,0	1,1,0
7	1,1,1	1,1,0	1,1,1	1,1,1	1,1,1	1,1,0	0,1,1	1,0,1	1,1,1

Table 4  
Example of an asynchronous evolution of a discrete system

$t$	State	$X$	$J(t)$	$s_1^{j(t)}(t)$	$s_2^{j(t)}(t)$	$s_3^{j(t)}(t)$	$X^s = \left( X_1^{s_1^{j(t)}(t)}, X_2^{s_2^{j(t)}(t)}, X_3^{s_3^{j(t)}(t)} \right)$	$X_{J(t)}^{t+1} = F_{J(t)}(X^s)$	$X^{t+1}$
0	4	1,0,0	2	0	0	0	1,0,0	1	1,1,0
1	6	1,1,0	2	1	1	1	1,1,0	0	1,0,0
2	4	1,0,0	3	1	1	0	1,1,0	1	1,0,1
3	5	1,0,1	1	3	3	3	1,0,1	0	0,0,1
4	1	0,0,1	2	1	0	2	1,0,0	1	0,1,1
5	3	0,1,1	1	3	1	0	1,1,0	1	1,1,1
6	7	1,1,1	:	:	:	:	:	:	:

Cartesian product set  $E = \prod_{i=1}^{\alpha} E_i$  and if the 3 conditions of Definition 2.1 are satisfied, then all asynchronous iterations from any initial state  $X^0 = (X_1^0, \dots, X_{\alpha}^0)$  converge to a unique fixed point  $X^*$  within  $p_x$  steps, i.e.

$$X^* = X^t, \quad t = p_x, p_x + 1, \dots,$$

where  $X^t$  is defined in Alg. (1) and  $\{p_l\}_{l \in \mathbb{N}}$ , in Definition 2.2.

The main steps of the proof are given in Appendix A.

### 3. Results

#### 3.1. Development of the research software DSE

The research software Discrete System Evolution (DSE) has been developed in order to analyse different convergence strategies of discrete models with asynchronous iterations. The discrete models studied are boolean neural networks with  $n$  neurons, i.e. each neuron at time  $t$  takes two states  $x_i^t \in \{0, 1\}$ . The network is partitioned into  $\alpha$  blocks, each block  $i$  having  $n_i$  components so that  $\sum_{i=1}^{\alpha} n_i = n$ . The state of the block  $i$  at time  $t$  is denoted by  $X_i^t$  and the state of the block system at time  $t$ , by  $X^t = (X_1^t, \dots, X_{\alpha}^t)$ . This software is based on four functionalities: the generation of a contraction matrix, the choice of different network parameters, the graphical representation of results and the statistical analysis of results.

(1) DSE allows the generation of different large ( $>100$ , the upper limit depending on the computer) boolean matrices  $B(f)$  satisfying the contraction hypothesis (with a spectral radius  $\rho(B) = 0$ ). Contraction matrices cannot easily be obtained with the available formal calculus software which does not include procedures for transforming matrices with boolean operators. A sample of different contraction matrices generated with a classical random function, allows statistical analysis (see point (4)).

(2) The different network parameters are:

(i) the size of the network: the number  $n$  of components ( $>100$ ), the number  $\alpha$  of blocks between 1 and  $n$ , the numbers  $n_i$  of components per block  $i$  which can be identical or different but with a sum equal to  $n$ .

(ii) the initial state  $X^0$  and the contraction matrix or the iteration function  $F$ .

(iii) the block delays in the set  $\{t - f(t), \dots, t\}$  verifying the condition  $\lim_{t \rightarrow \infty} s_j^i(t) = \infty$ , e.g.  $s_j^i(t)$  generated according to a uniform random distribution in  $\{t - t/20, \dots, t\}$ . If  $s_j^i(t) = t$  then the iterations are chaotic (particular case of asynchronous iterations).

(iv) the block strategies:

- Asynchronous block parallel (Jacobi) strategy:  $J(t) = \{1, \dots, \alpha\}$ , i.e. the  $\alpha$  blocks are iterated simultaneously.

$$\begin{aligned} \text{Given } X^0 &= (X_1^0, \dots, X_{\alpha}^0) \\ t &= 0, 1, \dots \\ i &= 1, \dots, \alpha \\ X_i^{t+1} &= F_i(X_1^{s_1^i(t)}, \dots, X_{\alpha}^{s_{\alpha}^i(t)}). \end{aligned} \tag{2}$$



- Asynchronous block series (Gauss–Seidel) strategy:  $J(t) = 1 + t \bmod \alpha$ , i.e. the  $\alpha$  blocks are iterated one after the other

$$\begin{aligned} &\text{Given } X^0 = (X_1^0, \dots, X_\alpha^0) \\ &t = 0, 1, \dots \\ &i = 1, \dots, \alpha \\ &X_i^{t+1} = \begin{cases} F_i(X_1^{s_i^t}, \dots, X_\alpha^{s_i^t}) & \text{if } i = 1 + t \bmod \alpha \\ X_i^t & \text{if } i \neq 1 + t \bmod \alpha. \end{cases} \end{aligned} \tag{3}$$

- Asynchronous block random strategy:  $J(t) \neq \emptyset$  and  $\forall i \in \{1, \dots, \alpha\}$ ,  $\text{Card}(\{t, i \in J(t)\}) = \infty$

$$\begin{aligned} &\text{Alg. (1) where the } \alpha \text{ blocks are iterated} \\ &\text{according to a uniform random distribution between 1 and } \alpha. \end{aligned} \tag{4}$$

- A particular case which we called “communication reassembly”: within each iteration  $t$ , the block  $i$  is independent from the others and its components are iterated in parallel mode  $p_i$  times. Then, before the time  $t + 1$ , all the results are communicated by a synchronization

$$\begin{aligned} &\text{Given } X^0 = (X_1^0, \dots, X_\alpha^0) \\ &t = 0, 1, \dots \\ &i = 1, \dots, \alpha \\ &Y_i^0 = X_i^t \\ &s = 0, \dots, p_i - 1 \\ &Y_i^{s+1} = F_i(X_1^t, \dots, Y_i^s, \dots, X_\alpha^t) \\ &X_i^{t+1} = Y_i^{p_i}. \end{aligned} \tag{5}$$

The block iteration parameter  $p_i$  generates delays between the blocks and the Alg. (5) is a particular case of the Alg. (1).

(3) The graphical functionality allows the display of results in a window. These curves are computed as follows. As the state  $x_i^t$  of each boolean neuron  $i$  is defined on the binary system  $E = \{0, 1\}$  of  $\text{Card}(E) = 2$ , the state  $x^t$  of the network with  $n$  neurons can be associated with the boolean number  $x_E^t = x_1^t \dots x_n^t$ . This number  $x_E^t$  can be represented on the decimal system  $\mathfrak{D} = \{0, \dots, 9\}$  as follows

$$x_{\mathfrak{D}}^t = \begin{cases} x_1^t 2^{n-1} + x_2^t 2^{n-2} + \dots + x_n^t & \text{if } n \leq 10, \\ x_1^t \times 2^{-1} + x_2^t \times 2^{-2} + \dots + x_n^t \times 2^{-n} & \text{if } n > 10. \end{cases}$$

If  $n \leq 10$ ,  $x_{\mathfrak{D}}^t$  is an integer number and exactly represented on a computer with values between 0 and 1023 ( $2^{10}$  possible values). If  $n > 10$ ,  $x_{\mathfrak{D}}^t$  is a real number and approximately represented on a computer with values in the range  $[0, 1]$ . The lower bound 0 is obtained when  $x_i^t = 0 \forall i \in \{1, \dots, n\}$  and the upper bound 1, when  $x_i^t = 1 \forall i \in \{1, \dots, n\}$  with  $n \rightarrow \infty$ . Therefore, these curves are represented as follows: the abscissa shows the time  $t$  by varying  $t$  between 0 and  $t_{\max}$ , and the ordinate gives the state of the network on the decimal system  $x_{\mathfrak{D}}$  with values either between 0 and 1023 (if  $n \leq 10$ ) or between 0 and 1 (if  $n > 10$ ). For large values of  $n$ , the curve cannot discriminate all the states of the network. For example, a network with  $n = 100$  neurons has  $2^{100} \approx 10^{30}$  possible states. Therefore, two kinds of information are given for

each  $t$ : the number  $x_{\mathfrak{D}}^t$  and a boolean indicator testing whether the state of the network at time  $t + 1$  is equal or not to the state at time  $t$  (by comparing the state of each neuron at the times  $t + 1$  and  $t$ ).

(4) The statistical functionality allows the generation of a sample of boolean neural networks having the same network parameters but different contraction matrices. Two classical statistical parameters are computed: the convergence mean time and its standard deviation.

Several utilities have also been developed: graphic tools (display of curves in a given range of  $t$  where the size, the lower and upper bounds are user parameters, the left/right scroll of curves), the printing of curves, the creation of numerical files, etc. Written in Pascal Delphi, this software can be implemented on microcomputers.

In summary, the research software DSE based on different network parameters allows the simulation of various discrete iteration processes.

### 3.2. Numerical applications

#### 3.2.1. A linear system

The numerical applications are based on a boolean neural network with  $n = 100$  neurons partitioned into  $\alpha = 10$  blocks according to the following way:  $n_1 = 11$ ,  $n_2 = 9$ ,  $n_3 = 13$ ,  $n_4 = 7$ ,  $n_5 = 15$ ,  $n_6 = 5$ ,  $n_7 = 17$ ,  $n_8 = 3$ ,  $n_9 = 19$ ,  $n_{10} = 1$ . The linear iteration function  $F$  chosen for the different simulations is  $FX = BX$ , where  $B$  is the contraction matrix. As  $B$  is a boolean contraction matrix, the unique fixed point from any initial state  $X^0$ , is  $X^* = (0, \dots, 0)$  associated with  $x_{\mathfrak{D}}^* = 0$ . The initial state chosen for the different simulations is the state with greatest distance to  $X^*$ , i.e.  $X^0 = (1, \dots, 1)$  associated with  $x_{\mathfrak{D}}^0 \simeq 1$ . Eight simulations with these network parameters have been investigated (Table 5):

Table 5  
Convergence mean time and its standard deviation of boolean neural networks with different types of strategies

Evolution	Convergence	
	Mean time	SD
<i>Asynchronous iterations</i>		
With a block parallel strategy: Alg. (2)	210	27
With a block series strategy: Alg. (3)	854	129
With a block random strategy: Alg. (4)	1576	270
<i>Communication reassembly: Alg. (5)</i>		
With a block iteration parameter $p_i = 1$	58	4
With a block iteration parameter $p_i = 2i$	53	4
<i>Chaotic iterations</i>		
With a block parallel strategy: Alg. (2) with $s_j^i(t) = t$	59	3
With a block series strategy: Alg. (3) with $s_j^i(t) = t$	352	25
With a block random strategy: Alg. (4) with $s_j^i(t) = t$	655	78

- three simulations with asynchronous iterations according to three block strategies, parallel: Alg. (2), series: Alg. (3), random: Alg. (4). The block delays are generated in the same way for these three simulations:  $s_j^i(t)$  has a uniform random distribution in  $\{t - t/20, \dots, t\}$ .
- two simulations with communication reassembly: Alg. (5) in function of the block iteration parameter  $p_i$ , one simulation with  $p_i = 1$ , i.e. each block is iterated one time, and one simulation with  $p_i = 2i$ , i.e. the number of iterations depends on the block  $i$ .
- three simulations representing particular cases of asynchronous iterations with  $s_j^i(t) = t$  (no delay), i.e. chaotic iterations according to the three previous block strategies: parallel, series and random.

For each simulation, a sample of 100 networks of 100 neurones (based on different contraction matrices) is constructed for computing the convergence mean time and its standard deviation. The statistical results obtained with this sample are significant and stable, e.g. a sample of 1000 networks leads to similar results for these two statistical parameters (data not shown).

Fig. 4 gives an example of convergence to  $X^*$  ( $x_D^* = 0$ ) of the network of 100 neurones with asynchronous iterations according to a block parallel strategy. Note that the network can reach the fixed point  $X^*$  several times before to stay on it.

The three types of simulations with asynchronous iterations and the two types of simulations with communication reassembly converge to a unique fixed point and are the first experimental results predicted by Theorem 2.1. In the case of asynchro-

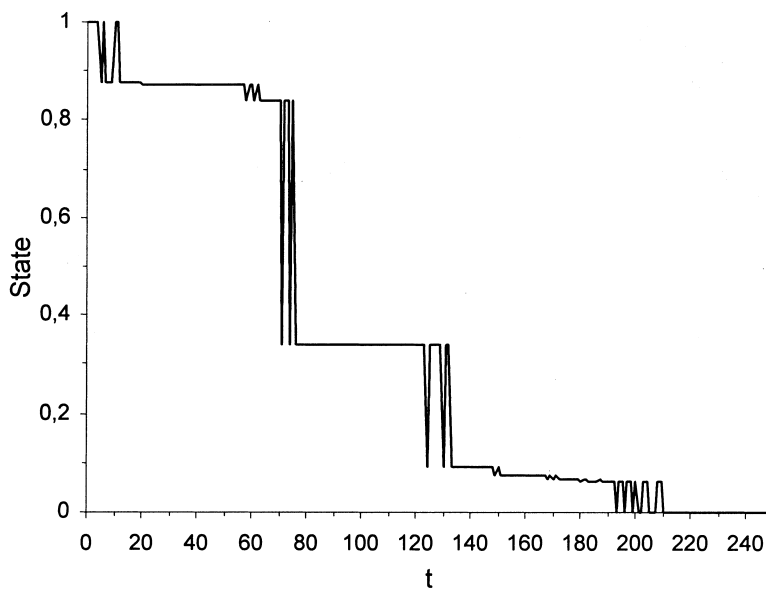


Fig. 4. Convergence of the boolean neural network with asynchronous iterations according to a block parallel strategy.

nous iterations, i.e. with unbounded delays, the mean times of convergence are, as expected, the lowest for the parallel strategy, medium for the series strategy and the highest for the random strategy (Table 5). Indeed, for the asynchronous parallel strategy, all blocks are updated at each  $t$  while for the asynchronous series strategy and the asynchronous random strategy, only one block is updated at each  $t$ . Furthermore, for the asynchronous series strategy, the blocks are updated one after the other leading to  $p_x$  steps (see Definition 2.2 and Theorem 2.1) shorter than the asynchronous random strategy where several identical blocks can be updated within the  $p_x$  steps. The same reasoning can be applied to the three types of strategies with the chaotic iterations leading to the expected observations in Table 5. On the other hand, for a given strategy, the convergence mean time with asynchronous iterations is higher than the one with chaotic iterations. Indeed, unbounded delays slow down the convergence process by considering previous states at some distance from the fixed point. Otherwise, the communication reassembly with a block iteration parameter  $p_i = 1$  converges to the fixed point with a mean time similar to the one of chaotic iterations with a parallel strategy.

3.2.2. *An application in the computer field leading to a nonlinear system*

Consider a computer system with  $n = 10$  processors  $P_i$ . Each processor computes a program which depends on some states  $x_i$  of some other processors  $P_j$ . This program is described according to the nonlinear iteration function  $f = (f_1, \dots, f_{10})$  defined as follows

$$f(x) = \begin{cases} f_1(x) = x_2x_4 + x_5x_6 + x_8\bar{x}_9 \\ f_2(x) = x_8 \\ f_3(x) = x_1 + \bar{x}_2 + \bar{x}_4\bar{x}_5 + x_7x_8x_9 \\ f_4(x) = x_2x_8x_9 \\ f_5(x) = x_2 + x_4x_8 + \bar{x}_9 \\ f_6(x) = x_2 + \bar{x}_4 + x_5 + x_8x_9 \\ f_7(x) = x_2(x_4 + x_5) + \bar{x}_6(x_8 + x_9) \\ f_8(x) = 1 \\ f_9(x) = \bar{x}_2 \\ f_{10}(x) = x_1(x_2x_3 + x_5) + \bar{x}_6\bar{x}_9. \end{cases}$$

For example, the processor  $P_6$  uses the states of the processors  $P_2, P_4, P_5, P_8$  and  $P_9$  (Fig. 5).

The problem associated with this example is the following one. What would be the behaviour of the computer under different types of evolution: parallel, series, block

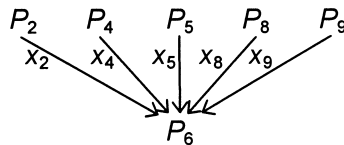


Fig. 5. Example of a relation between processors.

series, chaotic. Furthermore, what would be this evolution if there are delays due to the communications between the processors or due to the different sizes of computation.

The Theorem 2.1. asserts the stabilization of the processors to the solution  $x^*$  of the system  $x = f(x)$  from any initial state if the conditions of this theorem are satisfied.

The matrix  $B(f)$  associated with  $f$  is then deduced

$$B(f) = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

It can be verified without difficulty that  $B(f)$  is a contraction matrix as  $\rho(B) = 0$ .

Therefore, the asynchronous evolution of the computer will stabilize to the fixed point  $x^* = (1, 1, 1, 0, 1, 1, 1, 1, 0, 1)$  associated with  $x_E^* = 1110111101$  and represented on the decimal system by  $x_2^* = 957$  (graphical representation for  $n \leq 10$ ). The asynchronous componentwise iterations are based on random delays (satisfying the

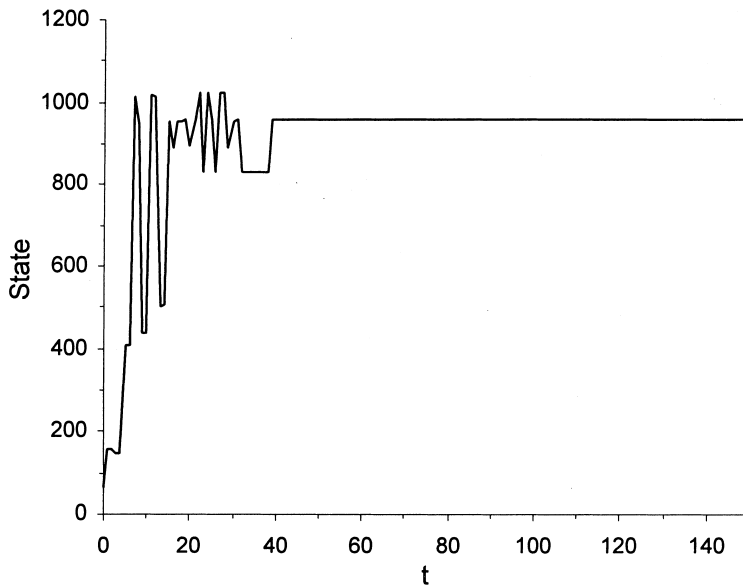


Fig. 6. Convergence of the computer system with asynchronous componentwise iterations and a random strategy.

condition (ii) of the Definition 2.1) and a random strategy. The initial state  $x^0$  chosen is the state with greatest distance to  $x^*$ , i.e.  $x^0 = (0, 0, 0, 1, 0, 0, 0, 0, 1, 0)$  associated with  $x_E^0 = 0001000010$  and  $x_D^0 = 66$ .

Fig. 6 shows the convergence to  $x^*$  of the computer system with random  $s_j^i(t) \in \{t/3, \dots, t\}$  and a random strategy  $J(t)$ .

#### 4. Conclusion

Different simulations of asynchronous evolution of discrete systems have been developed by varying the nature of the system, linear or nonlinear, its partition into blocks or not, the type of strategies and the delays. All these simulations converge and represent the first experimental results predicted by the theory. This work can be applied in several research fields such as parallel computing, neural networks, cellular automata, pattern recognition, transport, etc. The choice of the best strategy leading to the fastest convergence is an open theoretical problem, both in the continuous and discrete frameworks.

#### Acknowledgements

We thank Prof. F. Robert from the Institut National Polytechnique of Grenoble and the Referees for their advice.

#### Appendix A

We give the main steps of the proof of the convergence Theorem 2.1.

Denote: For  $t \in \mathbb{N}^*$  and  $i, j \in \{1, \dots, \alpha\}$

$$X^t = (X_1^t, \dots, X_\alpha^t),$$

$$X^{s^i(t)} = (X_1^{s_1^i(t)}, \dots, X_\alpha^{s_\alpha^i(t)}),$$

$$X^{s(t)} = (X^{s^1(t)}, \dots, X^{s^\alpha(t)}),$$

and  $A_{t-1}$ , a  $n \times n$  matrix defined by

$$(A_{t-1})_{ij} = \begin{cases} 1 & \text{if } i = j \notin J(t-1) \\ 0 & \text{if } i = j \in J(t-1) \\ 0 & \text{if } i \neq j. \end{cases} \quad (6)$$

The proof is divided into three parts.

(i)  $d(X^t, X^*) \leq A_{t-1}d(X^{t-1}, X^*) + B(F)d(X^{s(t-1)}, X^*)$ .

We first prove that

$$d(X^t, X^*) \leq A_{t-1}d(X^{t-1}, X^*) + d(F(X^{s(t-1)}), F(X^*)),$$

where for all integers  $t \geq 1$ ,  $A_{t-1}$  is defined by Eq. (6). By applying Proposition 1.1, we deduce

$$d(X^t, X^*) \leq A_{t-1}d(X^{t-1}, X^*) + B(F)d(X^{s(t-1)}, X^*).$$

- (ii)  $d(X^t, X^*) \leq \sum_{j=1}^q (B(F))^{k_j} d(X^0, X^*)$  with  $k_j \geq \alpha$  for  $t \geq p_\alpha$ .
- (iii)  $d(X^t, X^*) \leq \sum_{j=1}^q (B(F))^{k_j} d(X^0, X^*)$ .

We prove that there exists finite integers  $q$  and  $k_j$  such that

$$d(X^t, X^*) \leq \prod_{j=0}^{t-1} A_j d(X^0, X^*) + \sum_{j=1}^q (B(F))^{k_j} d(X^0, X^*).$$

Due to Definition 2.2 of  $\{p_l\}_{l \in \mathbb{N}}$ ,

$$\forall i \in \{1, \dots, \alpha\}, \forall l \in \mathbb{N}, \exists j \in \{p_l, \dots, p_{l+1} - 1\} \text{ so that } (A_j)_{ii} = 0.$$

Therefore,

$$\prod_{j=0}^{t-1} A_j = 0 \quad \text{if } t \geq p_1.$$

Then for  $t \geq p_\alpha$ ,

$$d(X^t, X^*) \leq \sum_{j=1}^q (B(F))^{k_j} d(X^0, X^*).$$

- (iib)  $k_j \geq \alpha$  for  $t \geq p_\alpha$ .

Due to the construction of  $\{p_l\}_{l \in \mathbb{N}}$  in Definition 2.2, if  $t \geq p_\alpha$  then  $k_j \geq \alpha$ . Indeed, all the blocks are updated at least  $\alpha$  times.

- (iii)  $\forall t \geq p_\alpha, d(X^t, X^*) = 0$ .

As  $F$  is a contraction, we have

$$\sum_{j=1}^q (B(F))^{k_j} = \sum_{j=1}^q (PLP^T)^{k_j} = P \sum_{j=1}^q L^{k_j} P^T,$$

where  $L$  is a strictly lower triangular block matrix. As  $PP^T = I, k_j \geq \alpha$  and as the product of  $\alpha$  strictly lower triangular block matrices of dimension  $\alpha$  leads to a null matrix, then for all  $j \in \{1, \dots, q\}$ ,

$$L^{k_j} = 0.$$

So

$$d(X^t, X^*) = 0.$$

In conclusion, the asynchronous discrete model converges to the unique fixed point  $X^*$  at most after  $p_\alpha$  steps.

## References

- [1] G.M. Baudet, Asynchronous iterative methods for multiprocessors, *J. ACM* 25 (1978) 226–244.

- [2] J.M. Bahi, C.J. Michel, Convergence of discrete asynchronous iterations, *International Journal of Computer Mathematics*, (1999) in press.
- [3] D.P. Bertsekas, J.N. Tsitsiklis, *Parallel and Distributed Computation*, Prentice Hall, Englewood Cliffs, 1989.
- [4] D. Chazan, W.L. Miranker, Chaotic relaxation, *Linear Alg. Appl.* 2 (1969) 199–222.
- [5] M.N. El Tarazi, Some convergence results for asynchronous algorithms, *Numer. Math* 39 (1982) 325–340.
- [6] J.-C. Miellou, Algorithmes de relaxation chaotique retard, *RAIRO R1* (1975) 55–82.
- [7] F. Robert, *Les Systèmes Dynamiques Discrets*, *Mathématiques and Applications* 19, Springer, Berlin, 1995.
- [8] F. Robert, M. Charnay, F. Musy, Itérations chaotiques série-parallèle pour des équations non-linéaires de point fixe, *Aplikace Matematiky* 20 (1975) 1–38.