

# Probabilistic Approach to the Asynchronous Iteration

A. S. Rasulov, M. T. Bakoev, D. R. Akabirhodjaeva

Mathematical Modeling and Informatics Department, University of World Economy and Diplomacy, Tashkent, Uzbekistan  
Email: [asrasulov@gmail.com](mailto:asrasulov@gmail.com)

Received October 2013

## ABSTRACT

In this work we will consider asynchronous iteration algorithms. As is well known in multiprocessor computers the parallel application of iterative methods often shows poor scaling and less optimal parallel efficiency. The ordinary iterative asynchronous method often has much better parallel efficiency as they almost never need to wait to communicate between processors. We will study probabilistic approach in asynchronous iteration algorithms and present a mathematical description of this computational process to the multiprocessor environment. The result of our simple numerical experiments shows a convergence and efficiency of asynchronous iterative processes for considered nonlinear problems.

## KEYWORDS

Asynchronous iterations; Linear and Nonlinear Equations; Multiprocessor Computers; Iterative Method with Memory; Probabilistic Approach

## 1. Introduction

Iterations methods for the solution linear and nonlinear equations are widely used because of their simplicity, fault tolerance, ease of parallelization. Historically, iterative algorithms were created and studied for using single processor computers. In multiprocessor computers the parallel application of iterative methods often shows poor scaling and less optimal parallel efficiency. As opposed to ordinary iterative asynchronous method often have much better parallel efficiency as they almost never need to wait to communicate between processors.

Our investigation concerns to study of asynchronous algorithms with probabilistic approach and presenting a mathematical description of this computational process to the multiprocessor environment.

## 2. Description of problem

Let  $X$  be a Banach space. In  $X$  we consider the following polynomial nonlinear equation:

$$\phi = \sum_{m=1}^{\infty} K_m(\phi, \dots, \phi) + f = P(\phi) \quad (1.1)$$

where  $f \in X, K_m : X^m \rightarrow X$  is a polynomial map,  $m = 1, 2, \dots, n$ , which is continuous and which is equivalent to their bounded-ness. Then there are exist constants  $C_m \geq 0$ , so that for any  $x_1, x_2, \dots, x_m \in X$

$$\|K_m(x_1, \dots, x_m)\| \leq C_m \prod_{i=1}^m \|x_i\|, \quad m = 1, n.$$

For the Equation (1.1) we create the following iterative method. Suppose that  $\varphi^{-n+2} = \varphi^{-n+3} = \dots = \varphi^0 = 0$  and at  $i \geq 1$  we define  $\varphi^{(i)}$  as the solution of the following equation

$$\varphi^{(i)} = f + K_1(\varphi^{(i)}) + \sum_{m=2}^n K_m(\varphi^{i-m+1}, \varphi^{i-m+2}, \dots, \varphi^i)$$

We will denote  $D_\delta^0(x) / D_\delta(x)$  as the open/closed balls in  $X$  with the radius  $\delta$  and centers at  $x$ . Let  $\mathbf{I}$  be the identity operator,  $L(X, X)$  is the space of linear continuous operators acting from  $X$  to  $X$ ,

$y = (y_1, \dots, y_{n-1}) \in X^{n-1}$ , and let

$z_y = z_{y_1, \dots, y_{n-1}}$  be equal

$$\text{to } I - K_1 - \sum_{m=2}^n K_m(y_{n-m+1}, y_{n-m+2}, \dots, y_{n-1}) \in L(X, X)$$

Suppose that  $(I - K_1)^{-1} \in L(X, X)$  exists. We will build a sequence  $(\varphi^i)_{i=n+2}^0 \subset D_\delta(0)$  in which

$$\varphi^{-n+2} = \dots = \varphi^0 = 0, \varphi^i = Z_{\varphi^{i-m+1}, \dots, \varphi^{i-1}}^{-1} f, i \geq 1. \tag{1.2}$$

**Lemma 1.** If the operator  $(I - K_1)^{-1}$  is defined and continuous, then there exists  $\delta, C, \theta \in (0, 1)$ , where the sequence (1.2) converges to  $\varphi^*$ , where  $\varphi^*$  is a solution of Equation (1.1) and has a minimal norm among all solutions from X to equation (1.1), and is unique in  $D_\delta(0)$ . Then the following inequality is true

$$\|\varphi^i - \varphi^*\| \leq C \frac{\theta^i}{1 - \theta} \|(I - K_1)^{-1} f\| \tag{1.3}$$

The proof of this lemma is given by [1].

### 2.1. Transformation Nonlinear Integral Equation into a System of Linear Integral Equations

Particularly, if

$$k_m(\varphi, \dots, \varphi)[x] = \int_D \dots \int_D K(x, y_1, \dots, y_m) \prod_{i=1}^m \varphi(y_i) dy_1 \dots dy_m,$$

$D \in R_s, x \in D, K_m$  is a given function in  $D^{m+1}, m = 1, \dots, n$ , then Equation (1.1) can be converted into an integral equation with a polynomial non-linearity, as considered in [2].

Let  $D^i$  be the direct product of i copies of D. Suppose  $x = x_1$ , multiply the obtained equation by  $\varphi(x_2)$ , and then again by  $\varphi(x_3)$  and so on, we denote  $\psi_i(x_1, \dots, x_i) = \prod_{j=1}^i \varphi(x_j)$  and we get the following infinite system of linear integral equations

$$\begin{aligned} \psi_i(x_1, \dots, x_i) &= \sum_{m=1}^{\infty} \int_D \dots \int_D K(x_1, y_1, \dots, y_m) * \\ &* \psi_{m+i-1}(y_1, \dots, y_m, x_2, \dots, x_i) dy_1 \dots dy_m + f(x_1) \psi_{i-1}(x_2, \dots, x_i), \psi_0 = 1, i = 1, 2, \dots \end{aligned} \tag{1.4}$$

Each function,  $\psi_i(x_1, \dots, x_i)$ , is defined on  $D^i$ . We will determine the operators  $K_i^{(m)}[\psi_{m+i}]$  in the following way

$$\begin{aligned} K_i^{(m)}[\psi_{m+i}] &= \int_D \dots \int_D K_m(x, y_1, \dots, y_m) \psi_{m+i}(y_1, \dots, y_m, x_1, \dots, x_i) dy_1 \dots dy_m \quad m = 2, 3, \dots \\ K_i^{(1)}[\psi_{1+i}] &= -\psi_{i+1}(x, x_1, \dots, x_i) + \int_D K_1(x, y) \psi_{1+i}(y, x_1, \dots, x_i) \\ K_i^{(0)}[\psi_i] &= f(x) \psi_i(x_1, \dots, x_i), \text{ (here } m = 0, 1) \end{aligned}$$

Now, we can formally Equation (1.1) with this infinite system of linear equations

$$\begin{aligned} K_i^{(n)}[\psi_{i+n}] + K_i^{(n-1)}[\psi_{i+n-1}] + \dots + K_i^{(1)}[\psi_{n+i}] + K_i^{(0)}[\psi_n] &= 0 \\ \psi_0 &\equiv 1, i = 0, 1, 2, \dots \end{aligned} \tag{1.5}$$

We consider closing the system (1.5) to some finite N, completing it with conditions

$$\psi_{N+n-1} = \psi_{N+n-2} = \dots = \psi_{N+1} = 0 \tag{1.6}$$

Let the domain D represent a simple interval [a,b]. We replace the integrals in system (1.5) with the cubature formulas using  $v^i$  mesh points, applied to each variable  $y_1, \dots, y_m (m = 1, \dots, n)$ . We then get a system with  $\sum_{i=1}^N v^i$  algebraic equations with the same number of unknowns. This system of linear algebraic equations has a specific structure. This is considered in detail in [1].

This system of linear Equation (1.4) will be rewritten in the following form

$$\begin{aligned} \psi_1 &= f_1(\psi_1, \dots, \psi_N) \\ \psi_2 &= f_2(\psi_1, \dots, \psi_N) \\ \psi_N &= f_N(\psi_1, \dots, \psi_N). \end{aligned} \tag{1.8}$$

We can write the system (1.8) in operator form as  $\psi = F(\psi)$ .

### 2.2. Convergence of the Asynchronous Method

Let's define [3,4] an asynchronous iterative method for solving the system (1.8). Let  $\{J_n\}_{n=1}^\infty$  be a sequence of non-empty subsets of set  $\{1, 2, \dots, N\}$  which is called chaotic sequence of sets.

Let the given initial vector be  $\psi(0)$ . We will construct the sequence  $\{\psi(n)\}_{n=1}^\infty$  of iterations by the following way [5-7]

$$\psi(n+1) = F_{J_{n+1}}(\psi(n)), n = 0, 1, 2 \tag{1.9}$$

Here

$$\psi_i(n+1) = \begin{cases} \psi_i(n), & \text{if } i \notin J_n \\ f_i(\psi(n)), & \text{if } i \in J_n, n = 0, 1, 2, \dots \end{cases} \tag{1.10}$$

The method of chaotic iterative (1.10) is a generalization of sequential iterative methods. If  $J_n = J = \{1, 2, \dots, N\}$  then all components of the iteration vector are updated at the same time and as a result we get the simple Jacobi iterative method  $\psi(n+1) = F(\psi(n)) n = 0, 1, 2, \dots$ .

The components of the iteration vector are refreshed the cyclically, if  $J_n = \{n \pmod N\}$ . In computing the following component we use previous one, which has already been computed using a Gauss-Seidel iteration. The main property of chaotic iterations is the random updating of the component iterative vector, which allows us to implement the efficiency on multiprocessor systems.

The generalization of method of chaotic iterative is the method of asynchronous iterations. The method of asynchronous iterations (1.8) is constructed with the following rule. Let  $\psi(0)$  be given, then

$$\psi_i(n) = \begin{cases} \psi_i(n-1), & \text{if } i \notin J_n \\ f_i(\psi_1(S_1(n)), \dots, \psi_N(S_N(n))), & \text{if } i \in J_n. \end{cases} \tag{1.11}$$

Here  $\psi_i$  is a component of the vector  $\psi$ , and  $\{S_i(n)\}_{n=1}^\infty, i = 1, N$  is a sequence of sets of non-negative integers, satisfying the following conditions, for any  $i = 1, N$ .

$$S_i(n) \leq n-1, n = 1, 2, \dots, i = 1, \dots, N \tag{1.12}$$

$$S_i(n) \rightarrow \infty, n \rightarrow \infty; i = 1, \dots, N \tag{1.13}$$

and every element  $i$  occurs infinitely many often in the sets  $J_n, n = 1, 2, \dots$ . The  $S_i(n), i = 1, N$  are called delays or lag. The condition (1.12) says that only components of previous iterates can be used in the evaluation of a new iterate. The condition (1.13) eventually says that values of an early iterate cannot be used any more in further evaluations, and more and more recent values of the components have to be used instead. The last condition "every element  $i$  occurs infinitely many often in the sets  $J_n, n = 1, 2, \dots$ " guarantees that no component is abandoned forever.

Let  $0 \leq S_i(n) < S_1$ , where  $S_1$  is the maximum number of iterations saved, *i.e.* when computing iterations we use components of vectors of previous iterative with no more than  $S_1$  in the past.

Let all elements in the sequence of subsets  $J_n$  have at least one element from  $\{1, 2, \dots, N\}$ . Then the following statement is valid

**Lemma 2.** For convergence of the asynchronous iterative method in  $R^n$  to the solution of (1.8) it is necessary that the spectral radius  $\rho(F) < 1$ .

The proof of this lemma is given in [8, p. 100]. This condition also provides convergence of a simple iterative process for the system (1.8). It is known that [2], the system (1.8) is equivalent to Equation (1.1) in the following sense. If the iterative method converges for (1.1) with initial value  $\varphi^{(0)} = f$ , then the method for (1.8) will converge, starting from the initial vector  $(f, 0, 0, \dots, 0)$ , and vice versa. Using the results of lemma 1 and 2 for the convergence of asynchronous iterations (1.11) the following assertion will be true [1].

**Theorem 1.** If the conditions of lemma (1.1) are satisfied, then the asynchronous iterative process (1.11) will

converge to the solution of the system (1.8).

The main difference between asynchronous iterative and the other iterative methods in parallel is the chaotic behavior of the vector components, which is expressed by the set of chaotic sequences  $J_n$ . The chaotic iterative process has the following two main advantages:

- 1) it is possible to calculate each coordinate of the iteration vector independently from the others (like the Monte Carlo method),
- 2) the convergence rate is higher, because this method sometimes essentially becomes an implicit iterative method like Gauss-Seidel method.

**Definition 1.** The sediment of a chaotic sequence  $\{J_n\}_{n=1}^\infty$  define as the set  $R = \{r \in \{1, 2, \dots, N\}, \forall n \exists m \geq n : r \in J_m\}$ .

**Definition 2.** The chaotic sequence of set has maximal sediment if  $R = \{1, 2, \dots, N\}$ .

Let consider the equation  $\psi = F(\psi)$ , where  $F$  is a nonlinear operator.

**Definition 3.** Let  $D(F)$  be the domain of the operator  $F$ . We shall call the operator  $F$  a  $p$ -Lipschitzian contraction in  $D(F)$  if there exists on  $N \times N$  matrix  $L$  with non-negative elements, satisfying the following inequality  $\forall y, z \in D(F); p(F(y) - F(z)) \leq Lp(y - z)$ .

The matrix  $L$  is called a Lipschitzian matrix for the operator  $F$ .

Let  $\{J_n\}_{n=1}^\infty$  be the sequence of nonempty subsets of set  $\bar{N} = \{1, \dots, N\}$ . Each element of  $J_n$  is generated via a distribution from the set  $\bar{N}$ . Let the elements of  $J_n$  be generated in a following way: the element  $i \in \bar{N}$  occurs in  $J_n$  with probability  $p_1$ . The probability of absence of  $i \in \bar{N}$  in  $J_n$  is equal to  $p_2$ , where  $p_2 = 1 - p_1$ . The average number of occurrences  $i \in \bar{N}$  in a sequence  $\{J_n\}_{n=1}^\infty$  can be computed. The average number of occurrences in  $n$  steps is equal to  $n p_1$ . Every element  $i \in \bar{N}$  thus occurs infinitely often in the sets  $J_n$  as  $n \rightarrow \infty$ . Thus we can formulate the following statement without proof.

**Lemma 3.** If the probability of occurrence of an arbitrary element  $i$  in  $J_n$  obeys the Bernoulli law of distribution with probability of success  $0 < p_1 < 1$ , or obeys the Poisson law, then the chaotic sequence  $\{J_n\}_{n=1}^\infty$  will have maximal sediment.

**Theorem 2.** Suppose the operator  $F: X \rightarrow X$  is a  $p$ -contraction on  $X$  with contraction matrix  $L$  and the conditions of lemma 1.3 are satisfied. Then for any  $y(0) \in X$ , the sequence of iterations, built up by the asynchronous method (1.11), converges to the unique fixed point of operator  $F$  in  $X$ .

**Proof.** In the  $p$ -contraction of the operator  $F$  on  $X$ , there is a unique fixed point  $\xi$  and  $p(F(y) - F(z)) \leq L p(y - z)$ . Since  $L$  is contraction matrix, there is positive vector  $v \in R^n$  and positive scalar  $\theta \in (0, 1)$ , such that  $Lv \leq \theta v$ . Let  $y(0)$  be an initial approximation. There exists an  $\alpha > 0$ , and  $n \geq n_q$ , where  $n_q$  is a iterative number, so that we can build a sequence  $\{J_n\}_{n=1}^\infty$  where the following will be true

$$p(y(0) - \xi) \leq \alpha \theta^q v, \tag{1.13a}$$

Then from inequality (1.13a) and taking the fact that  $\theta < 1$ , it follows that  $p(y(n)) \rightarrow \xi$  given that  $q \rightarrow \infty$  i.e.  $\|y_i(n)\| \rightarrow \xi_i$  when  $n \rightarrow \infty, i = 1, \dots, N$ . We will show, assuming  $q = 0$ , that  $p(y(n) - \xi) < \alpha v, n > n_0 = 0$ . We use mathematical induction. We assume it true for  $n = k - 1$ . From the definition of the asynchronous iterative process, it follows that components of the vector  $y(k)$  are given as  $y_i(k) = y_i(k - 1)$ , if  $i \notin J_k$  and than  $\|y_i(k) - \xi_i\| = \|y_i(k - 1) - \xi_i\| < \alpha v_i$ . The equation  $y_i(k) = f_i(y_i(S_i(k)))$  holds if  $i \in J_k$ , since  $S_i(k) < k - 1$ , then  $\|f_i(y(S_i(k))) - \xi_i\| = \|y_i(k - 1) - \xi_i\| \leq \alpha \theta v_i$ . Since  $\theta < 1$ , the last inequality we say  $\|y_i(k) - \xi_i\| \leq \alpha v_i$ , and therefore  $p(y(n) - \xi) \leq \alpha v$ .

Let's suppose that iteration numbers  $n_1, n_2, \dots, n_{q-1}$  are known and condition (1.13a) is satisfied. Let's find  $n_q$  for which condition (1.13a) is satisfied. We first determine a number  $r = \min\{k: \forall n \geq k, S_i(n) \geq n_{q-1}, i = 1, \dots, N\}$ . Since  $S_i(n) \rightarrow \infty$ , if  $n \rightarrow \infty$ , then there exists an  $r$ , such that  $p(y(r) - \xi) \leq \alpha \theta^{q-1}$ . Let  $n > r$ , from the  $p$ -contraction property of the operator  $F$ , we get

$$p(F(z) - \xi) < L(p(z) - \xi) \leq \alpha \theta^{q-1} L v \leq \alpha \theta^q v$$

Let  $n_q = \min\{n : n > r, J_r \cup \dots \cup J_n = \{1, 2, \dots, N\}\}$ .

If  $i \in J_n$  then  $\|y_i(n) - \xi_i\| = \|f_i(y_i(S_i(n))) - \xi_i\| \leq \alpha \theta^q v_i$  and this holds for all  $i$ , if  $n \geq n_q$ . From the definition  $n_q$  it follows that all components of iteration vector  $y(n_q)$  differ from the component of the iteration vector  $y(r)$ , and the theorem has been proven.

There are various methods for implementing asynchronous iterations.

To conclude this section, we provide the following asynchronous iterative method with memory.

Suppose  $\psi(0), \dots, \psi(m-1)$  are known. The sequence of vectors  $\{z_i^r(n)\}_{n=1}^\infty, i = 1, \dots, N$ , are given in the fol-

lowing expressions

$z_i^r(n) = \psi_i(S_i^r(n))$ , and the  $\{S_1^1(n), \dots, S_N^1(n), S_1^2(n), \dots, S_N^m(n)\}$   $n = m, m + 1, \dots$  are sequence of vectors from  $N^{N \times m}$ .

The following conditions hold

$$\begin{aligned} \max\{S_i^r(n) : 1 \leq r \leq m\} &\leq n - 1, n = m, m + 1, \dots \text{ and} \\ \min\{S_i^r(n) : 1 \leq r \leq m\} &\rightarrow \infty \text{ as } n \rightarrow \infty. \end{aligned}$$

**Definition 4.** The iterative process  $\{\psi(n)\}_{n=1}^\infty$  is defined as

$$\psi_i(n) = \begin{cases} \psi_i(n-1), & \text{if } i \notin J_n \\ f_i(z_1^r(n), \dots, z_N^r(n)), & \text{if } i \in J_n, \end{cases}$$

is called a asynchronous iterative method with memory.

It is known [9] that if the operator  $F$  is a  $p$ -contraction operator and the chaotic sequence  $\{J_n\}$  had maximal sediment, then the asynchronous iterative process with memory converges to the unique fixed point of the operator  $F$ .

The asynchronous iterative method with memory size  $m > 2$  has a problem. The method requires saving a very large number of items in memory. However, this kind of asynchronous method is used in special cases. So, for example, let us use the method Newton-Kantorovich, for solving the equation  $\psi = F(\psi)$ . This method is defined by formula

$$\begin{aligned} \psi(n) &= H(\psi(n-1), \psi(k(n))), k(n) \leq n - 1 \text{ and} \\ H(y, z) &= y - [F'(y)]^{-1} F(z) \quad n = 1, 2, \dots \end{aligned}$$

This iterative process is not the same as the asynchronous iterative processes (1.11). To compute the current iterative, we use the result of two previous iterations. In general case, if an iterative process of  $m$ -th order is used, *i.e.*

$\psi(n+1) = G_n(\psi(n), \psi(n-1), \dots, \psi(n-m+1))$  then we can't use an asynchronous method (1.11). In this case, we must use an asynchronous iterative method with memory  $m$ .

To estimate the speed of convergence of the iterative method we usually use the value [6]

$$R = \liminf_{n \rightarrow \infty} \frac{-\log \|\psi(n) - \xi\|_0}{n}, \tag{1.14}$$

where  $\xi$  is a fixed point of the operator  $F$ , and  $\|\psi\|_0 = \max_{i=1, N} \|\psi_i\|$ .

An efficiency of an iterative method we usually use the value:

$$E = \liminf_{n \rightarrow \infty} \frac{-\log \|\psi(n) - \xi\|_0}{c_n} \tag{1.15}$$

where,  $c_n$  is the cost associated with the evaluation of the first  $n$  iterates. Usually we choose  $c_n$ , proportional to the number of arithmetic operations and are necessary to compute the first  $n$  iterations, or the computer time used for computing of  $n$  iterations. We note that if  $c_n/n$  tends to some finite  $\tau$  (which corresponds to the average cost per step), then the efficiency is simple given by  $E = R/\tau$ .

In the asynchronous iteration case, we can determine the value of  $c_n$  in the following way:

$c_n = \frac{|J_1| + \dots + |J_n|}{n}$ , where  $|J_i|$  is the cardinality of the set  $J_n$ , *i.e.* number of components evaluated at the  $n$ -th step of the iterative. In this case, the cost is better suited for a parallel implementation, and can be evaluated through the classical tools of queuing theory. Let's denote the number of macro iterations (accumulated iterative) in the asynchronous process with as  $q_n$ . Then we obtain

$$R \geq -\lim_{n \rightarrow \infty} \left(\frac{q_n}{n}\right) \log \rho(L), \quad E \geq -\lim_{n \rightarrow \infty} \left(\frac{q_n}{c_n}\right) \log \rho(L),$$

where  $L$  is a contraction matrix of the operator  $F$ .

The results obtained in this section will be applied to solving a real life problem in the next section.

### 3. Probabilistic Model for the Asynchronous Algorithms

#### 3.1. Asynchronous Algorithms Bernoulli and Poisson Distributions

In this section we give another way of organizing asynchronous computations for solving equations. Suppose a computer has  $S$  processors. The first  $l_1$  components of system (1.8) will be solved on the first processor, and the second  $l_2$  components of the system (1.8) on the second processor etc. Each of the  $S$  processors computes its own group of components of the unknown vector  $\psi$  simultaneously. At the end of iteration the value of group components are copied to computer memory of and they become accessible to all processors. This procedure works for shared memory multiprocessors.

In this case, asynchronous calculations can be represented in the following form:

$$\psi_i^T = q_i^T f_i(\psi^{T_i}) + (1 - q_i^T) \psi_i^T, \quad i = 1, S \tag{2.1}$$

where  $\psi_i^T (i = 1, S)$  is an element of a vector of the current approximation at time  $T$ .

$\psi_i^T = (0, 0, \dots, \psi_i^T, 0, \dots, 0)$ ,  
 $\psi^{T_i} = (\psi_1^{T_i}, \psi_2^{T_i}, \dots, \psi_S^{T_i})$  are vectors of the current approximation which are calculated at time  $T_i < T$ . Here

$T_i$  denotes the beginning of the next calculation step of the group component  $\psi_i^T$ .  $f_i(\psi^{T_i}) = \delta_i F(\psi^{T_i})$ , where  $\delta_i$  is a diagonal matrix, the components  $\delta_{ij}$  are equal to one if component  $x_j$  is calculated by the  $i$ -th processor, otherwise they are zero. The coefficients  $q_i^T, i = 1, S$  will be set to one when the time interval  $T - T_i$  is long enough to calculate and write the vector  $\psi_i^T$  into shared memory, of computer and otherwise it will be set to zero.

The time interval  $T - T_i$  depends on the time  $T$ , and  $T_i$  denotes the beginning of the next time step for calculating group components.  $T$  could be a random variable that has a specific distribution.

Let's divide  $T$  into time intervals. For  $T_i \in T$  the equality  $T^i = T - n_i^T$  is valid, where,  $n_i^T = T - T^i$  is an integer depending on  $T$  and  $n_i^T \geq 1$ . Let  $n_j^T > n_i^T$ , thus numbers  $T - n_j^T$  denote the starting time of the next time step, and the end of the previous step. Let  $p_i^T$  be an integer that takes values of either  $n_i^T$ , or  $n_j^T$ . For  $\psi^T = \psi_1^T + \dots + \psi_S^T$  we will transform Equation (2.1) to the form of

$$\psi^T = \sum_{i=1}^S f(x^{T-p_i^T}) \tag{2.2}$$

Equation (2.2) gives a new method of organizing this asynchronous computational process for the solving Equation (1.8). Analyzing and comparing these two asynchronous methods (2.2) and (1.11, 1.12, 1.13), we can conclude their equivalence. Indeed, in (2.2) we use values of the other components  $\psi^T$  to compute component  $f_i(x^{T-p_i^T})$ , which were calculated up to time  $T - p_i^T$  i.e. the most recent values. This meets conditions (1.12). In (2.2), the number of iterative of each component of  $\psi_i(T - p_i^T)$  tends to infinity, when  $T \rightarrow \infty$ , which agrees with condition (1.13). In a asynchronous iterative process (1.11), actual task is manipulate the sequences  $\{J_n\}_{n=1}^\infty$  and  $\{S_i(n)\}_{n=1}^\infty$ . But for real computations in multiprocessor and parallel systems, these sequences are unknown and they are determined implicitly by a process.

This is essentially the difficulty of studying asynchronous computations. As we assumed in the beginning of this section and in this case, the sequences  $J_n$  and  $S_i(n)$  can be considered random with a specific distribution. In [10], a new result was obtained for convergence conditions for asynchronous computations in case (2.2).

We will now consider the asynchronous process (1.11) in more detail. We feel that probabilistic model describes the practical situation much better. We suppose, for example, a set  $J_j$  is randomly chooses from the set  $\{1, \dots, N\}$ . Further, we assume that the computation time for the next iterative for each processors is nondeterministic. The time computation is an independent random variable, and has some simple distribution.

At first, suppose  $F$  is a linear operator. Let's close system of linear algebraic Equations (1.5 - 1.6). This system is then written in the following form  $\psi = A\psi + b$ , where  $A$  is an  $(N \times N)$  matrix. Suppose there are  $m$  parallel processors. We will assume that for a random time interval, each processor is able to compute the  $n$ -th iterative with probability  $p$ , and is not able to compute in with probability  $q = 1 - p$ . When it cannot compute in time, on the next iteration step, we use a known component according to (1.11), since  $S_i(j) = j - i$ .

Let  $\|\cdot\|$  represent some norm, and  $\psi = \lim_{k \rightarrow \infty} \psi(k) = (I - A)^{-1} b$ .

**Theorem 3.** Suppose our processors behave consistent to the Bernoulli distribution. If a vector  $\xi(n)$  is the

random value of the n-th iteration and  $\|A\| < 1$ , then  $\lim_{n \rightarrow \infty} E \xi(n) = \bar{\psi}$ , where  $\bar{\psi}$  is the solution of the problem.

**Proof.** It's easy to see that the probability of reaching the n-th full iteration is  $p^n$ . The probability  $P_n(k)$ , of reaching the n-th step, with  $k$  successful iterations is equal to  $P_n(k) = \binom{n}{k} p^k q^{n-k}$ .

Let's now compute the expected value of the random variable,  $\xi(n)$ . Because  $\|A\| < 1$ , we have

$$\psi(k) = (I + A + \dots + A^k) b = (I - A)^{-1} (I - A^{k+1}) b \tag{2.4}$$

Then, we obtain  $E \xi(n) = (I - A)^{-1} b + (I - A)^{-1} A (Iq + Ap)^n b$ . Here we see that  $\lim_{n \rightarrow \infty} E \xi(n) = \bar{\psi}$  because  $\|A\| < 1$ , and this proves the theorem.

Now consider the case when the time is distributed according to a Poisson distribution with the parameter  $\lambda = np$ , where  $n$  is the number of iterative and  $p$  is the probability of a successful computation in the next iteration step.

Let's denote

$$\xi(n) = \left( \begin{matrix} \psi(0) \dots \psi(k) \dots \psi(n) \\ p(0, \lambda) \dots p(k, \lambda) \dots p(n, \lambda) \end{matrix} \right),$$

where  $p(k, \lambda) = \frac{\lambda^k}{k!} e^{-\lambda}$ .

In this case the following theorem is valid:

**Theorem 4.** Suppose our processors behave in accordance to the Poisson distribution. If a vector  $\xi(n)$  is the random value of the iteration at the in n-th, step and  $\|A\| < 1$ , then  $\lim_{n \rightarrow \infty} E \xi(n) = \bar{\psi}$ .

**Proof.** Using (2.4) we have that the expected value of  $\xi(n)$  is

$$\begin{aligned} E \xi(n) &= \sum_{k=0}^n e^{-\lambda} \frac{\lambda^k}{k!} \psi(k) = (I - A)^{-1} e^{-\lambda} \sum_{k=0}^n (I - A^{k+1}) \frac{\lambda^k}{k!} b = \\ &= (I - A)^{-1} (I - A e^{-\lambda(E-A)}) b \square = (I - A)^{-1} (I - A e^{-np(E-A)}) b \end{aligned}$$

with a fixed  $p$ , we obtain  $\lim_{n \rightarrow \infty} E \xi(n) = \bar{\psi}$  and this proves the theorem.

Now let's look at the nonlinear case for  $\psi = F \psi$  (2.5)

Here  $F$  is a nonlinear  $p$ - Lipschitzian operator contraction on  $X$  with contraction matrix  $L$ . Suppose that we have  $N$  ( $N$  is a number of equations) processors, each with probability  $p$  successfully computing the next iteration for its components of  $\psi_k(n)$ . Let a sequence of random variables,  $J_n$ , have a Bernoulli distribution. If  $p \neq 0$ , it will tend to have the maximum sediment. According to theorem 2 of the previous section, we can assert that the asynchronous iterative method converges to the unique fixed point,  $\xi$ , of the operator  $F$ .

Let  $S_i(n) = n - 1$ . Since  $L$  is a contraction matrix, then for  $\alpha > 0$  and  $\theta \in (0, 1)$  it is known that there exists a positive vector,  $v \in R^N$ , such that  $l(\psi(n) - \xi) < \alpha \theta v$  when  $n > n_q$ . Here  $n_q = \min\{n : n \geq r, J_1 \cup J_2 \cup \dots \cup J_n = \{1, 2, \dots, N\}\}$ . This means that all components of the iteration vector,  $\psi(n_q)$ , differ from the components of iterative vector  $\psi(r)$ .

$\|\psi\|_v$  is defined as  $\|\psi\|_v = \max_{1 \leq i \leq N} \frac{\|\psi_i\|}{v_i}$ . We choose  $\alpha = \|\psi(0) - \xi\|_v$ . Let  $q_n$  be the maximal number of macro-iterative in the sequence of iterative  $\{\psi(n)\}_{n=0}^\infty$ . Then

$$\|\psi(n) - \xi\|_v = \alpha \theta^{q_n}, \quad n = 0, 1, 2, \dots \tag{2.6}$$

In this case  $q_n = np$  i.e.  $q_n$  is the average number of successes in n-th step of iteration.

Let  $R$  be the rate of convergence. Then  $R$ , from (1.14) give the following inequality  $R \geq -p \log(\rho(L)) \approx -\lim_{n \rightarrow \infty} p \log \theta$ , since it's known that  $\theta$  can be chosen as close to  $\rho(L)$  as possible.

We recall that in case of the simple iterative method,  $R \geq -\log \rho(L)$ .

In the case when  $S_i(n) = n - 1$ , a constant  $c_n$  characterizing the average complexity of the computation for  $n$  asynchronous iteration can be determined by  $c_n = \frac{|J_1| + \dots + |J_n|}{n}$ , where  $|J_i|$  is cardinality of the set  $J_i$ . The efficiency of the asynchronous iterative process in this case is determined from (1.15) and

$E \geq -\lim_{n \rightarrow \infty} \left( \frac{np}{c_n} \right) \log \rho(L)$ . Thus, since the probability of successes  $p \leq 1$ , the asynchronous iterative method

converges to the fixed point slower than the simple Jacobi iterative method, however the efficiency of asynchronous iterative process is better.

### 3.2. Numerical Experiment

To illustrate our points we solved the following nonlinear boundary value problem in the unit square:

$$\begin{aligned} \Delta u(x, y) &= u^2(x, y) + 4 - (x^2 + y^2)^2 \\ u(x, y) \Big|_{x=0} &= y^2, \quad u(x, y) \Big|_{y=0} = x^2, \quad u(x, y) \Big|_{x=1} = 1 + y^2, \\ u(x, y) \Big|_{y=1} &= 1 + x^2. \end{aligned}$$

The exact solution of this problem is  $u(x, y) = x^2 + y^2$ . Using the finite difference method, we get the following system of nonlinear algebraic equations:

$$u_{ij}^{(k)} = \frac{1}{4} (u_{i+1,j}^{(k-1)} + u_{i-1,j}^{(k-1)} + u_{i,j-1}^{(k-1)} + u_{i,j+1}^{(k-1)}) - \frac{h^2}{4} \times ((u_{ij}^{(k-1)})^2 + 4), \quad i, j, = \overline{1, n}, \quad k = 1, 2, \dots$$

Here  $k$  is the step of iteration number. Suppose we use five processor systems, due to the five point stencil we use. With probability  $p$  each processor is able to compute the next iteration in time. The results of the numerical experiments are given in the table below, where  $p$  is the probability of success. Here  $n$  is the number of iterations,  $T$  is the relative unit CPU time,  $\tilde{u}$  is iterative and  $\mathbf{u}$  is the exact solution.

n-the number of iterations	T-CPU time	p	$\tilde{u}$ (0.7,0.9)-iterative solution	$\mathbf{u}$ (0.7,0.9)-exact solution
100	1	1	1.26	1.30
200	2	0.5	1.27	1.30
300	1	0.6	1.16	1.30

The numerical experiments were done, with  $\{J_n\}_{n=1}^{\infty}$  and  $\{S_i(n)\}_{n \rightarrow \infty}^{\infty}$  having a different distributions.

### 4. Conclusions

The asynchronous iterative method converges to the fixed point slower than the simple Jacobi iterative methods, however the computational efficiency of asynchronous iterative process is better.

The results of our simple numerical experiments show the convergence and efficiency of asynchronous iterative processes for considered nonlinear problems.

### REFERENCES

- [1] A. S. Rasulov, "Asynchronous and Statistical Algorithms for the Solution Linear and Nonlinear Equations," PhD Dissertation, Novosibirsk, 1992, Russia.  
<http://fizmathim.com/asinhronno-statisticheskie-algoritmy-resheniya-lineynyh-i-nelineynyh-uravneniy>
- [2] S. M. Ermakov, V. V. Nekrutkin and A. S. Sipin, "Random Processes for Classical Equations of Mathematical Physics," Kluwer Academic Publication, London, 1989.
- [3] G. M. Baudet, "Iterative Methods for Asynchronous Multiprocessors, High Speed Computer and Algorithm Organization," In: D. Kuck, D. Lawrie and A. Sameh, Eds., Academic Press, New York, 1977, pp. 309-310.
- [4] G. M. Baudet, "Asynchronous Iterative Methods for Multiprocessors," *Journal of Associated Computation*, Vol. 25. No. 2, 1978, pp. 226-244. <http://dx.doi.org/10.1145/322063.322067>
- [5] F. Robert, M. Charnay and F. Musy, "Iteration Chaotiques Serie-Paralele Pour Equations Non-Linearires de Point Fixe," *Applied Mathematics*, Vol. 20, No. 1, 1975, pp. 1-38.
- [6] D. Chazan and W. Miranker, "Chaotic Relaxation," *Linear Algebra and Its Applications*, Vol. 2, 1969, pp. 199-229. [http://dx.doi.org/10.1016/0024-3795\(69\)90028-7](http://dx.doi.org/10.1016/0024-3795(69)90028-7)
- [7] J. D. Donnelly, "Periodic Chotic Relaxation," *Linear Algebra and Its Applications*, No. 2, 1971, pp. 117-128. [http://dx.doi.org/10.1016/0024-3795\(71\)90033-4](http://dx.doi.org/10.1016/0024-3795(71)90033-4)
- [8] B. B. Nesterenko and V. A. Marchuk, "Basics of Asynchronous Methods for Parallel Calculations," Kiev. Nauka-Dumka, 1989, p. 176, Monograph (in Russian).



- [9] W. Shooman, "Orthogonal Processing, Parallel Processors, System, Technologies and Application," 1970, pp. 297-308.
- [10] V. N. Belectiy, "Multiprocessors and Parallel Structures with Organization of Asynchronous Calculations," Kiev, Nauka-Dumka, 1988, p. 240 (in Russian).