

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ АСИНХРОННЫХ  
КЛЕТОЧНО-АВТОМАТНЫХ АЛГОРИТМОВ

О.Л.Бандман

*Институт вычислительной математики и математической геофизики СО РАН  
Г.Новосибирск*e-mail: [bandman@ssd.sccc.ru](mailto:bandman@ssd.sccc.ru)

Обсуждается проблема распараллеливания алгоритмов клеточно-автоматного моделирования процессов на атомно-молекулярном уровне. Поскольку эти процессы требуют большого времени моделирования, возникает необходимость параллельной реализации вычислений, что для асинхронных клеточных автоматов является сложной задачей. В статье предложен способ преобразования асинхронного клеточного автомата к эквивалентному блочно-синхронному, что существенно улучшает эффективность распараллеливания.

**Ключевые слова:** математическое моделирование, клеточный автомат, асинхронные взаимодействия, параллельная реализация, синхронно-асинхронные преобразования

**1. Введение**

Клеточный автомат (КА), введенный полвека тому назад фон-Нейманом для иллюстрации возможности построения машины, имитирующей биологические явления, получает в наши дни второе дыхание как математическая модель пространственной динамики. Основные свойства КА – естественная мелкозернистая параллельность и локальность взаимодействий – делает их привлекательными для создания вычислительных технологий, соответствующих современным и быстро развивающимся многомашинным и многопроцессорным вычислительным комплексам. Предполагается, что КА-подход к моделированию пространственной динамики станет дополнением к дифференциальным уравнениям в частных производных [1]. Активизации интереса к КА-моделированию способствуют два фактора, характеризующие современное состояние вычислительной науки: технический и математический. Технический фактор обусловлен развитием многопроцессорных кластеров и компьютерных систем, которые требуют простых и эффективных технологий распараллеливания задач больших размеров, что легко достигается благодаря мелкозернистости КА-алгоритмов. Математический фактор связан с трудностями, с которыми сталкиваются традиционные вычислительные методы, особенно в части эффективного распараллеливания. Кроме того, существует и физическое обоснование правомерности КА-моделирования. Например, в [2] указывается на то, что Ньютонова механика описывает только обратимые процессы, тогда как для многих природных процессов характерна необратимость. С увеличением интереса к биологии и физике выявилось много явлений, для которых дифференциальных уравнений вообще не существует. Отсюда возникло желание вернуться к истокам построения математических моделей, т.е. к описанию явлений на микроскопическом уровне, оперируя дискретными перемещениями реальных или абстрактных частиц, иногда наделяя их скоростью движения в дискретном пространстве. Такие представления явлений легко выражаются в терминах КА с простыми детерминированными или вероятностными правилами переходов.

В настоящее время существует много клеточно-автоматных моделей, различающихся алфавитом состояний, структурой дискретного пространства и так называемыми «режимами» функционирования. Классический КА фон-Неймана имеет булев алфавит состояний, детерминированные правила переходов и синхронный режим функционирования. Такие КА могут моделировать диффузионные, волновые процессы, фазовые переходы [2]. Они хорошо изучены, легко программируются и распараллеливаются на любое число процессоров с высокой эффективностью (обычно >90%) без деградации. Химические реакции [4], физические процессы на молекулярно и атомарном уровне [5] моделируются вероятностными асинхронными КА, которые в своих областях знаний обычно называются «методами Монте-Карло». Очевидно, что этот класс КА требует больших вычислительных мощностей, поскольку для выявления каких-либо свойств моделируемых процессов необходимо оперировать очень

большими количествами частиц, и, следовательно, КА должен иметь очень большие размеры ( $10^7 - 10^9$ ). Поскольку на одном процессоре (даже самом быстродействующем) не удастся моделировать КА таких размеров за приемлемое время, очень важно уметь разместить КА на многих процессорах. К сожалению, опыта параллельных реализаций асинхронных КА пока не существует, тогда как трудности видны явным образом. Они связаны с тем, что для обеспечения корректности необходимо межпроцессорные взаимодействия производить при каждом изменении состояния на границах частей КА. Возникают вопросы: какой эффективности параллельной реализации можно добиться и как сделать ее приемлемой. Эти вопросы поднимаются в этой работе и намечаются возможные пути поиска ответов на них.

Статья состоит из трех разделов. Сразу после введения дается формальная постановка задачи. Во втором разделе обсуждается возможность преобразования КА к синхронному виду. Третий раздел посвящен методам распараллеливания.

## 2. Формальное определение асинхронного КА-алгоритма.

Клеточный автомат, который теперь называется классическим, обычно определяется множеством клеток, плотно заполняющих дискретное  $D$ -мерное пространство. Расстояния между центрами соседних клеток одинаковы и равны 1. Каждой клетке ставится в соответствие конечный автомат, называемый элементарным, входами которого являются выходы других клеток, находящихся в некотором ограниченном соседстве. Все элементарные автоматы одинаковы. Они имеют булев алфавит состояний и меняют свои состояния все одновременно, синхронно переводя КА в следующее глобальное состояние. Итеративный процесс перехода КА из состояния в состояние называется эволюцией.

Интересующий нас асинхронный клеточный автомат (далее просто КА) отличается от классического двумя свойствами. Во-первых, правила переходов элементарных автоматов из состояния в состояние – вероятностные. Во-вторых, режим его функционирования – асинхронный, т.е. каждый элементарный автомат совершает смену состояний в случайный момент времени. Поскольку все элементарные автоматы одинаковы, можно считать, что промежутки времени между двумя сменами состояний у них одинаковы. Это позволяет предположить, что глобальные смены состояний КА происходят через равные промежутки времени. Далее они называются итерациями, а последовательность итераций в процессе функционирования – эволюцией КА.

Формально КА рассматриваемого типа может быть представлен той же тройкой понятий, что и классический КА: алфавитом  $A$ , множеством имен элементарных автоматов  $M$ , и набором локальных операторов  $\Phi$ ,  $CA = \langle A, M, \Phi \rangle$ , и указанием на асинхронный режим применения локальных операторов. Алфавитом является конечное множество целых неотрицательных чисел, в частности, это может быть булево множество. Множество имен – конечное множество наборов координат дискретного пространства  $M = \{ m : m = (i, j, k), i = 0, 1, \dots, I, j = 0, \dots, J, k = 0, \dots, K \}$ . Конкретные значения  $A$  и  $M$  определяют класс клеточных массивов  $\Omega(A \times M)$ , каждый представитель которого  $\Omega = \{(a, m) : a \in A, m \in M\}$  представлен множеством пар вида  $(a, m)$ , называемых клетками. Клетке соответствует элементарный автомат с именем  $m \in M$  и состоянием  $a \in A$ , причем состояние может иметь смысл переменной с областью определения  $A$ . Переменные состояния далее обозначаются символами  $v, u, w$  в отличие от констант  $a, b, c \in A$ . На основном множестве имен определены именуемые функции:  $\varphi : M \rightarrow M$ . Если  $m_k = \varphi(m)$ , то клетка с именем  $m_k$  является соседом клетки  $m$ . Конечное множество именуемых функций составляет шаблон соседства вида

$$T(m) = \{j_0(m), \dots, j_q(m)\}, \quad (1)$$

в котором  $j_0(m) = m$ . В свою очередь шаблон соседства определяет в клеточном массиве группу клеток-соседей клетки  $m \in M$  образует локальную конфигурацию

$$S(m) = \{(v_0, m), \dots, (v_q, j_q(m))\}. \quad (2)$$

которое называется локальной конфигурацией с определяющим шаблоном  $T(m)$ . Клетка  $(v_0(m))$  в локальной конфигурации называется центральной.

Три локальных конфигурации, записанные в виде подстановки

$$\Theta(m) : S(m) \rightarrow S'(m), \quad (3)$$

составляют локальный оператор. Определяющие шаблоны для  $S(m)$  и  $S'(m)$  в (3) находятся в соотношении в следующем соотношении:

$$T'(m) \subseteq T(m), \quad T'(m) = \{(m), \dots, j_p(m)\}, \quad p \leq q \quad (4)$$

Состояния конфигурации правой части (3)

$$S'(m) = \{(f_0, m), \dots, (f_q, j_q(m))\} \quad (5)$$

являются значениями функций от переменных состояния всех клеток конфигурации  $S(m)$ , т.е.

$$f_k = f(v_0, \dots, v_q), \quad (6)$$

Локальная конфигурация  $S(m)$  совместима с клеточным массивом  $\Omega \in \Omega(A, M)$ , что обозначается как  $S(m) \subseteq \Omega$ , если

- переменные состояний  $v_k$  во всех клетках  $S(m)$  определены на алфавите  $A$ ,
- клетки с константными состояниями  $(a_k, j_k(m)) \in S(m)$  входят также в  $\Omega$ .

Локальный оператор  $\Theta(m)$  вида (3) применим к  $\Omega \in \Omega(A, M)$ , если найдется хотя бы одна клетка  $m \in M$ , для которой выполняется условия:  $S(m) \subseteq \Omega$ .

Применение  $\Theta(m)$  к  $\Omega$  состоит в замене состояний в клетках  $(v_k, j_k(m)) \in \Omega$  на значения функций  $f_k$ , вычисленных по (5).

### 3. Синхронный и асинхронный КА

Существует два основных режима применения локальных операторов для вычисления следующего глобального состояния  $\Omega(t+1)$ : синхронный и асинхронный. При синхронном режиме аргументами функции переходов (5) служат состояния клеток  $(a_i, m) \in \Omega(t)$  на текущей итерации. На каждой итерации каждая клетка вычисляет значения функций переходов (6), а замена состояний на новые производится только тогда, когда все клетки выполняют эти вычисления. Таким образом, имитируется *синхронная параллельная* смена состояний. Естественно, что при работе на одном процессоре эти свойства являются «виртуальными».

При асинхронном режиме каждая клетка вычисляет функцию перехода в новое состояние от тех значений состояний соседей, которые на данный момент имеют место, и сразу меняет свое состояние, т.е. аргументы функций переходов (6) могут быть состояниями клеток как из  $\Omega(t)$ , так и из  $\Omega(t+1)$ . Таким образом, при асинхронном режиме каждая итерация состоит из  $|\Omega|$  последовательных микрошагов (тактов), на каждом из которых только одна выбранная случайным образом клетка изменяет свое состояние.

Названные два режима (синхронный и асинхронный) являются двумя крайними случаями. Существует много алгоритмов и со смешанными и производными режимами. Например, синхронно-блочный [6] и асинхронный упорядоченный [7].

Различные значения аргументов в функциях переходов при разных режимах приводят к тому важному факту, что при одном и том же наборе операторов и одном и том же исходном клеточном массиве, эволюции синхронного и асинхронного КА совершенно различны. При этом, формального способа построения синхронного КА по заданному асинхронному не существует. Хотя, известно что существуют КА, которые при обоих режимах приходят к одному и тому устойчивому состоянию [8]. Более того, известна теорема о том, что для каждого синхронного КА можно построить такой асинхронный, который будет моделировать эволюцию исходного с растяжкой во времени. К сожалению, обратное не доказано. А именно этот факт было бы полезно знать, так как это помогло бы решить проблемы параллельной реализации асинхронных КА на многопроцессорных машинах без общей памяти. Следующий пример показывает, насколько могут быть разными эволюции КА, отличающиеся только режимами функционирования.

**Пример 1. КА-модель распространения волны** – это первая и самая простая модель газовой динамики, получившая названия НРР-модели по первым буквам своих трех авторов [9]. Газ представлен клеточным массивом  $\Omega = \{v, (i, j)\}$  с алфавитом состояний  $A = \{v: v = (s_1, s_2, s_3, s_4), s_i \in \{0, 1\}\}$ , содержащим булевы векторы длиной 4. Множество имен  $M = \{(i, j) : i, j = 0, \dots, N\}$ . Значение  $s_l = 1$  в векторе состояния клетки с именем  $(i, j)$  означает, что в соответствующей точке моделируемого пространства находится «модельная» частица, движущаяся с единичной скоростью по направлению к  $l$ -му соседу. Соседи клетки  $(i, j)$  пронумерованы в соответствии с порядком именуемых функций в шаблоне  $T(i, j) = \{(i, j), (i-1, j), (i, j+1), (i+1, j), (i, j-1)\}$ . Режим функционирования этой модели синхронный, двухтактный, т.е. каждая итерация делится на два такта, называемых *фазой сдвига* и *фазой тактом столкновения*. На фазе сдвига каждая частица перемещается в соседнюю клетку, на которую указывает ее вектор скорости. На фазе столкновения происходит смена направления движения на  $90^\circ$  тех пар частиц, которые попали в одну и ту же клетку, двигаясь навстречу друг другу. Набор локальных операторов записывается в следующем виде.

$$\begin{aligned} \Theta_1(i, j): \{(s_l, (i, j)), (s'_l, j_l(i, j))\} &\rightarrow \{(s'_l, (i, j))\}, & l=1, 2, 3, 4; \\ \Theta_2(i, j): \{(0, 1, 0, 1), (i, j)\} &\rightarrow \{(1, 0, 1, 0), (i, j)\}, \\ \Theta_3(i, j): \{(1, 0, 1, 0), (i, j)\} &\rightarrow \{(0, 1, 0, 1), (i, j)\}; \end{aligned} \quad (7)$$

На рис. 1 показано 3 глобальных состояния в эволюции этого КА. Видно, что при начальном состоянии в виде плотного квадрата в центре приводит к появлению круговой волны, распространяющейся от центра.

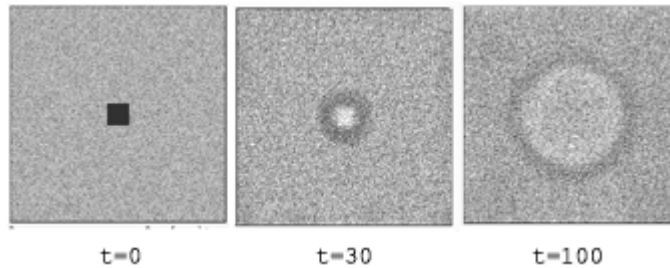


Рис. 1. Синхронное функционирование модели НРР-газ: распространение круговой волны в газовой среде. Глобальные состояния при  $t=0$ ,  $t=10$ ,  $t=50$  итераций .

При асинхронном функционировании КА с теми же локальными операторами, но применяемыми на каждой микроитерации к случайно выбранной точке при тех же начальных данных получается совсем другая эволюция ( рис.2.).

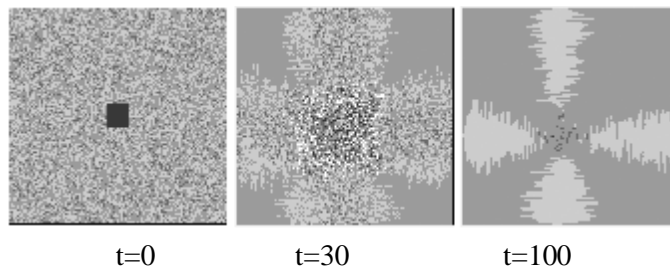


Рис.2. Асинхронное функционирование модели НРР-газ: глобальные состояния при  $t=0$ ,  $t=100$ ,  $t=320$

#### 4. Проблема параллельной реализации асинхронных КА.

Поскольку асинхронные КА используются, в основном, для моделирования дискретных сред, имитируя движения или превращения реальных или «модельных» (стилизованых под реальные) атомов и молекул, то вследствие малости размеров последних требуются большие размеры клеточных пространств. Кроме того, из-за стохастического характера поведения среды, требуется большое время моделирования. Отсюда возникает необходимость реализации КА алгоритма на многопроцессорных системах. В настоящее время многопроцессорные системы различаются на два класса: кластеры объединяющие много (от десятков до 1000) компьютеров и многоядерные компьютеры с общей памятью, объединяющие несколько (4-16) процессоров, которые затем тоже соединяются в кластер. При размещении задачи на кластере в параллельных программах используется библиотека MPI, при использовании системы с общей памятью – библиотека OpenMP. Во всех случаях пересылки данных с компьютера на компьютер необходимы.

Время пересылки пакета данных  $t_{\text{обмена}} = t_{\text{лат}} + Vt_{\text{бит}}$ , причем  $t_{\text{лат}} \gg t_{\text{бит}}$ , где  $t_{\text{лат}}$  – время латентности,  $t_{\text{бит}}$  – время передачи бита,  $V$  – число передаваемых бит. Отсюда следует, что пересылать данные большими пакетами выгоднее. Поэтому, при синхронных обменах эффективность параллельной реализации  $Q = T_{\text{моно}} / nT_{\text{multi}}$  близка к единице, даже если она выполняется с использованием заблокированных операций, при которых легко выполняются условия безопасности и отсутствия дедлоков. Важно только чтобы в каждом процессоре выполнялось условие

$$t_{\text{счета}} \gg t_{\text{обмена}} \quad (8)$$

где  $t_{\text{счета}}$  и  $t_{\text{обмена}}$  – времена счета и обмена в каждом процессоре, соответственно.

Поскольку при асинхронном режиме необходимо выполнять обмены на каждой микроитерации, на которой изменяются состояния в смежных клетках (лежащих на границах раздела массива по процессорам), то большой эффективности достигнуть не удастся, т.к. на каждую передачу состояний нескольких (обычно двух) смежных клеток затрачивается полное время  $t_{\text{обмена}}$ . Поэтому условие (8) оказывается здесь более жестким, чем в синхронном случае. Пусть размеры клеточных подмассивов  $|M_k|$  на всех процессорах одинаковы и равны  $L \times L$ , тогда число смежных клеток равно  $4L$ . Если время выполнения локальных операторов перехода в следующую итерацию равно  $\tau$ , то условие (8) для асинхронного распараллеливания примет вид

$$L^2\tau \gg 4L(t_{\text{лат}} + 2t_{\text{бит}}) \quad (9)$$

Применение неблокированных операций MPI, которые допускают выполнение счета во время обмена, может позволить улучшить ситуацию. Однако, частые и перекрестные передачи требуют еще немалых усилий и дополнительных затраты времени, чтобы избежать дедлоков. Наш первый опыт параллельной реализации задачи эпитаксиального роста кремниевого кристалла на кластере ССКЦ СО РАН показали, что при  $L^2 = 10^4$  ускорение в 4 раза можно получить только на 12 процессорах, что, естественно, не удовлетворяет исследователей, побуждая к поиску иных путей.

### 5. Синхронно-блочный режим функционирования КА

Одним из путей повышения эффективности распараллеливания асинхронных КА является поиск преобразования асинхронного КА в квазисинхронный с эволюцией близкой к исходному асинхронному.

Идея такого преобразования состоит в следующем. Пусть определяющий шаблон локального оператора  $T(i,j)$  имеет радиус  $R$ . Это значит, что во всех его именуемых функциях вида  $\varphi(i,j)=(i-a,j-b)$  константы  $a \in R, b \in R$ . Тогда на клеточном массиве следует построить  $r=(2R+1)^D$  ( $D$ -размерность массива) покрытий  $\pi_0, \dots, \pi_k, \dots, \pi_{r-1}$  клеточного массива  $\Omega$  состоящих их одинаковых блоков, т.е.  $\pi_k = \{B_{k0}, \dots, B_{k\mu}\}, \mu = |\Omega|/r, \forall k=0, \dots, r-1$

$$B_{k\alpha} \cap B_{k\beta} = \emptyset \quad \forall \alpha, \beta \in \{0, \dots, \mu\}, \quad \text{и} \quad \bigcup_{\alpha=0}^{\mu} B_{k\alpha} = \Omega \quad (10)$$

Каждый блок содержит локальную конфигурацию. В каждом покрытии все блоки равны. От покрытия к покрытию они различаются только расположением в блоке центральной клетки. По времени каждая итерация подразделяется на  $r$  последовательных тактов. На каждом  $k$ -м такте локальные операторы применяются к центральным клеткам блоков  $k$ -го покрытия. Поскольку блоки не пересекаются, результат не зависит от порядка их применения и от того, делалось это последовательно или одновременно. Предполагается, что генератор случайных чисел работает идеально, т.е. на каждой итерации в каждой клетке происходит одно и только одно применение локального оператора. Если это не так, то эволюции могут незначительно различаться, однако при длительной работе КА их можно считать статистически равными. Таким образом, если на  $0$ -м шаге  $t$ -й итерации все функции переходов используют переменные состояний вида  $v(t), u(t)$ , то на первом шаге  $r$ -ая часть всех клеток массива имеет состояния  $v(t+1)$  и  $u(t+1)$ , а после  $(r-1)$ -го шага массив примет глобальное состояние  $\Omega(t+1)$ . Поскольку применения локальных операторов на каждом шаге абсолютно разделены по пространству, то порядок выбора клеток для их применения (случайный или нет) не влияет на результат в конце такта.

**Пример 2.** Преобразование асинхронного КА в блочно-синхронный проще всего показать на простом примере одномерной «наивной» диффузии [6], который представлен булевым алфавитом состояний, множеством имен  $M=\{i:=0,1,\dots,N-1\}$  и двумя вероятностными локальными операторами:

$$\begin{aligned} \Theta_1(i): \{(u,i-1), (v,i), (w,i+1)\} &\rightarrow \{(v,i-1), (u,i), (w,i+1)\} \text{ при } rand < 0.5, \\ \Theta_2(i): \{(u,i-1), (v,i), (w,i+1)\} &\rightarrow \{(u,i-1), (w,i), (v,i+1)\} \text{ при } rand \geq 0.5, \end{aligned} \quad (11)$$

в которых  $i=\text{rand} \times N$  вычисляется на каждом асинхронном такте. Синхронное применение этого оператора вообще невозможно, так как оператор противоречив. Это значит, что применение его, например, к подмассиву  $\{(1,i-1), (0,i), (0,i+1)\}$  оставит неопределенным состояние клетки  $i$ , так как  $\Theta(i-1)$  стремится его изменить на 1, тогда как  $\Theta(i)$  должна оставить его равным 0 (оператор применим только к паре клеток с разными состояниями). Радиус шаблона здесь  $R=1$ , следовательно  $r=3$ . Итерация подразделяется на три такта  $\tau_0, \tau_1, \tau_2$ , в каждом из которых выполняются локальные операторы (11), причем  $\tau_0 = \tau \pmod{3}$ ,  $\tau_1 = \tau \pmod{3} + 1$ ,  $\tau_2 = \tau \pmod{3} + 2$ , где  $\tau$  - микрошаг асинхронного режима.

В целом, блочно-синхронный режим выделяет из всего множества трасс асинхронного процесса некоторое их подмножество. Насколько серьезно это отклоняет нас от истинного положения вещей можно узнать из опыта моделирования. Небogatый пока опыт сравнительных исследований асинхронных и блочно-синхронных КА существенных отклонений не выявил, что показано на следующем примере.

**Пример 3.** Процесс окисления монооксида кремния на катализаторе. На рис 3 показаны

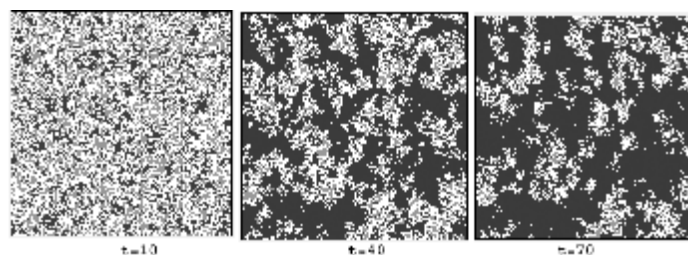


Рис.3. Три глобальных состояния при асинхронном (Монте-Карло) моделировании процесса окисления СО на катализаторе  
глобальные состояния КА при асинхронном, а на рис.4 при блочно-синхронном моделировании одного и того же процесса окисления монооксида углерода [4] на каталитической поверхности

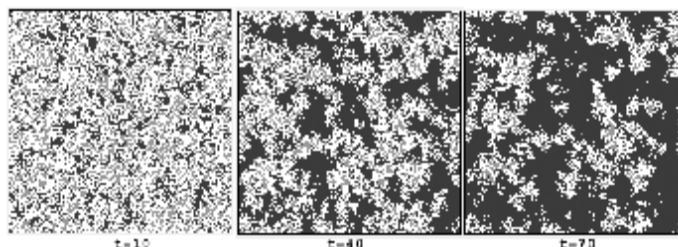


Рис.3 Три глобальных состояния при блочно-синхронном режиме моделировании процесса окисления СО на катализаторе

Кроме того, что блочно-синхронный алгоритм хорошо распараллеливается, он и в последовательной реализации дает ускорение вычислений на 20-30% по сравнению с синхронным, а время распараллеливания при этом сокращается существенно. Условие эффективного распараллеливания принимает следующий вид

$$L^2\tau \gg r(L_{\text{лат}} + 4L_{\text{пер}}), \quad (11)$$

что с учетом (8) позволили получить при  $L^2=10^4$  эффективность параллельной реализации 70-80 %.

#### Литература.

1. *Toffoli T.* Cellular Automata as an Alternative to (rather than Approximation) to Differential Equations in Modeling Physics // *Physica D.* – Vol.10, 1984. – 117-127
2. *Wolfram S.* Statistical mechanics of Cellular Automata // *Review of Modern Physics.* - Vol.55. – 1993. - 607-640
3. *Wolfram S.* A new kind of Science – Wolfram Media Inc., Champaign, Ill., USA, 2002
4. *Latkin E.I., Elokhin V.I., Gorodetski V.V.* Spiral concentration waves in the MonteCarlo model of CO oxidation over Pd(110) caused by synchronization via  $\text{CO}_{\text{ads}}$  diffusion between separate parts of catalytic surface // *Chemical Engineering Journal.* – Vol.91, 2003. – 123-131.
5. *Succi S.* Kinetic Approach to Lattice Quantum Mechanics // <http://www.iac.rm.cnr.it>
6. *Торфолли Т., Марголюс Н.* Машины клеточных автоматов М. : Наука . 1988.
7. *Park J.K., Steiglitz K., Thurston W.P.* Soliton-like behavior in automata // *Physica D.* - Vol. 19, 1986. – 423-432.
8. *Ачасова С.М., Бандман О.Л.* Корректность параллельных вычислительных процессов – Новосибирск: Наука – 1998.
9. *Hardy J., Pomeau Y., de Pazzis O.* Time evolution of two-dimensional model system // *Journal of Math. Physics.* – Vol.14, 1973. – 1746-1759

Бандман Ольга Леонидовна, г.н.с. ИВМиМГ СО РАН, д.т.н. проф., тел. (8 383) 330 89 94

