

УДК 519.6

## МЕТОДЫ РАСПАРАЛЛЕЛИВАНИЯ ВЫЧИСЛЕНИЙ ПРИ РЕШЕНИИ УРАВНЕНИЯ ПЕРЕНОСА НЕЙТРОНОВ НА НЕСТРУКТУРИРОВАННЫХ СЕТКАХ В ПРОГРАММЕ "РАДУГА Т"

О. В. Николаева\*  
(ИПМ им. М. В. Келдыша РАН, г. Москва)

Рассматривается задача распараллеливания вычислений при решении интегродифференциального уравнения переноса на неструктурированных сетках. Распараллеливание выполняется разбиением пространственной сетки на подобласти с расчетом каждой подобласти на своем MPI-процессе. Рассмотрены два метода распараллеливания вычислений, включенных в программу "Радуга Т": ВJ (Block-Jacobi) и BS (Block-Seidel). В ВJ-методе выполняются простые итерации по интегралу столкновений и по значениям решения на границах пространственных подобластей. В BS-методе, следуя идее известного КВА (Koch-Baker-Alcouffe) метода, простые итерации выполняются только по интегралу столкновений. Таким образом, в BS-методе число простых итераций меньше, чем в ВJ-методе, но их выполнение требует большего времени. Выполнено исследование эффективности ВJ- и BS-методов распараллеливания вычислений при решении модельной задачи по программе "Радуга Т".

*Ключевые слова:* уравнение переноса нейтронов, неструктурированные сетки, параллельные алгоритмы, Block-Jacobi-метод, Block-Seidel-метод, КВА-метод.

### Введение

Решение интегродифференциального уравнения переноса нейтронов в активных зонах и защитных слоях ядерных реакторов требует выполнения большого объема вычислений. С одной стороны, искомая функция распределения нейтронов зависит от шести переменных (три пространственные и три скоростные координаты). С другой стороны, размер области расчета может достигать 50–100 м<sup>3</sup>. Поэтому при решении таких задач активно используются параллельные вычисления. Так как решаемое уравнение интегродифференциальное, для решения системы сеточных уравнений применяется метод простой итерации.

Распараллеливание вычислений наиболее часто выполняется по пространственным переменным; при этом в памяти каждого процессора хранятся только величины, относящиеся к соответствующей пространственной подобласти. Значения решения на границах подобласти в ВJ (Block-Jacobi) методе находятся расчетом в соседних подобластях на предыдущей простой итерации. Таким образом, решение, полученное в ВJ-методе на каждой простой итерации, различно при различном числе подобластей. С ростом числа подобластей растет и число необходимых для сходимости процесса простых итераций.

В связи с этим был развит КВА (Koch-Baker-Alcouffe) метод, в котором на каждой простой итерации выполняется сквозной по всей области расчет, когда значения решения на границе подобласти (функции распределения входящих в подобласть нейтронов) находятся расчетом в соседней подобласти на той же простой итерации (это функция распределения выходящих из соседней подобласти

\* Николаева Ольга Васильевна, старший научный сотрудник,  
e-mail: nika@kiam.ru

нейтронов). В КВА-методе получаемое на каждой простой итерации решение, а также число необходимых для сходимости процесса простых итераций не зависят от числа подобластей. Такой эффект достигается разбиением каждой простой итерации на несколько шагов. На каждом шаге расчет выполняется только для некоторой части ячеек подобласти для некоторых направлений вектора скорости; далее полученные значения решения передаются в соседнюю подобласть и используются там для расчета на следующем шаге.

В КВА-методе неизбежны простые MPI-процессы, дожидаящихся получения от соседних MPI-процессов необходимых для расчета значений решения. Кроме того, в КВА-методе требуется выполнять больше MPI-обменов, чем в VJ-методе. Поэтому в значительной части компьютерных программ применяется VJ-метод как на структурированных [1], так и на неструктурированных сетках [2, 3]. В программе ARES [4] для структурированных сеток используется КВА-метод, а для неструктурированных — VJ-метод.

КВА-метод первоначально был развит для структурированных сеток. Разработаны и разрабатываются новые алгоритмы определения порядка расчета пространственных ячеек для уменьшения времени простоев MPI-процессов [5, 6]. Для дальнейшего сокращения времени расчета используются графические ускорители [7, 8], применяется конвейеризация вычислений [9–11]. В работе [12] получено оптимальное разбиение сетки на подобласти в зависимости от числа ядер, в [13] применяется динамическое разбиение на подобласти. Дополнительное распараллеливание по скоростям [8–10, 14, 15] позволяет использовать больше процессорных ядер при том же числе пространственных подобластей.

КВА-метод, разработанный для структурированных сеток, может быть легко обобщен на неструктурированные сетки, если границы подобластей параллельны координатным плоскостям.

В некоторых случаях удастся построить неструктурированную сетку с плоскими границами подобластей. Например, в [16]  $xy$ -сечение области расчета разбивается на подобласти-прямоугольники и в каждой подобласти строится треугольная сетка, которая далее преобразуется в сетку из треугольных призм. Для уменьшения неизбежно возникающего здесь дисбаланса числа ячеек в подобластях выполняется специальная процедура коррекции сетки. Однако даже после этой процедуры дисбаланс может достигать 10 %.

В некоторых сеточных схемах каждая пространственная ячейка разбивается на подъячейки плоскостями, параллельными вектору скорости нейтрона, и в каждой подъячейке сеточные уравнения строятся независимо (так называемый SBA (Slice Balance Approach) метод). Тогда для каждого вектора скорости можно построить разбиение сетки на подобласти с границами, параллельными этому вектору [17]. Такой подход позволяет сократить простые, однако требует хранения всей сетки в памяти каждого MPI-процесса и обменов значениями решения в граничных ячейках.

Для схем на неструктурированных сетках, не использующих SBA-подход, для уменьшения времени простоев MPI-процессов разрабатываются достаточно сложные алгоритмы определения порядка расчета пространственных ячеек [18–20]. При этом необходимо, чтобы ячейки, результат расчета которых определяет функцию распределения нейтронов, выходящих из данной подобласти в соседнюю, обрабатывались ранее, чем ячейки, результат расчета которых влияет на функцию распределения нейтронов, покидающих данную подобласть через внешнюю границу всей области.

Первоначально порядок расчета ячеек определялся предварительно [18]. Позднее были созданы методы, в которых он определяется динамически, в процессе счета [19, 20]; предварительно задается только правило нахождения приоритета для каждой ячейки. Отдельно рассматривается вопрос о выборе способа обмена данными между MPI-процессами — с буферизацией или без нее [20].

В настоящей работе для расчета на неструктурированных сетках предлагается BS (Block-Seidel) метод. Как и КВА, этот метод на каждой простой итерации реализует сквозной счет по всей области. BS-метод включен в программу "Радуга Т" [21], которая содержит также и VJ-метод. В настоящей работе выполнено сравнение эффективности распараллеливания обоих методов при решении модельной задачи.

## 1. Задача для уравнения переноса нейтронов

Рассмотрим уравнение переноса нейтронов с фиксированной кинетической энергией

$$\boldsymbol{\Omega} \nabla \Psi(\mathbf{r}, \boldsymbol{\Omega}) + \sigma(\mathbf{r}) \Psi(\mathbf{r}, \boldsymbol{\Omega}) = \int_{\Omega} \Psi(\mathbf{r}, \boldsymbol{\Omega}') \sigma_s(\mathbf{r}, \boldsymbol{\Omega} \cdot \boldsymbol{\Omega}') d\boldsymbol{\Omega}' + Q(\mathbf{r}, \boldsymbol{\Omega}). \quad (1)$$

Здесь искомая функция  $\Psi(\mathbf{r}, \boldsymbol{\Omega}) [1/(\text{см}^2 \cdot \text{с} \cdot \text{ср})]$ , плотность потока нейтронов, определяет число нейтронов в точке  $\mathbf{r} = (x, y, z)$  трехмерной выпуклой области  $G$  с единичным вектором скорости  $\boldsymbol{\Omega} \in \Omega$ ,  $\Omega$  — единичная сфера;  $\sigma [1/\text{см}]$  — полное сечение взаимодействия нейтрона со средой;  $\sigma_s(\mathbf{r}, \boldsymbol{\Omega}, \boldsymbol{\Omega}') [1/(\text{см} \cdot \text{ср})]$  — сечение рассеяния;  $Q(\mathbf{r}, \boldsymbol{\Omega}) [1/(\text{см}^3 \cdot \text{с} \cdot \text{ср})]$  — плотность источника нейтронов.

Зададим для уравнения (1) следующие краевые условия:

$$\Psi(\mathbf{r}, \boldsymbol{\Omega}) = f(\Psi(\mathbf{r}, \boldsymbol{\Omega})) \quad \text{при} \quad \boldsymbol{\Omega} \cdot \mathbf{n}(\mathbf{r}) < 0, \quad \mathbf{r} \in \partial G. \quad (2)$$

Здесь  $\mathbf{n}(\mathbf{r})$  — внешняя нормаль к границе  $\partial G$  области  $G$  в точке  $\mathbf{r}$ ;  $f [1/(\text{см}^2 \cdot \text{с} \cdot \text{ср})]$  — плотность потока граничного источника. Важной в приложениях функцией является плотность скалярного потока нейтронов  $\Phi(\mathbf{r}, \boldsymbol{\Omega}) [1/(\text{см}^2 \cdot \text{с})]$ :

$$\Phi(\mathbf{r}, \boldsymbol{\Omega}) = \int_{\Omega} \Psi(\mathbf{r}, \boldsymbol{\Omega}') d\boldsymbol{\Omega}'.$$

Разобьем сферу  $\Omega$  на четыре квадранта, в каждом квадранте зададим набор из  $K$  векторов. Всего имеем  $4K$  векторов:  $\{\boldsymbol{\Omega}_{q,k}, q = 1, \dots, 4, k = 1, \dots, K\}$ .

Введем в области  $G$  неструктурированную пространственную сетку, состоящую из  $J$  ячеек — выпуклых многогранников. Введем сеточные величины, например, значения решения в вершинах ячеек и в вершинах граней ячеек для каждого из векторов  $\boldsymbol{\Omega}_{q,k}$ . Грани ячеек, аппроксимирующие границу  $\partial G$ , будем называть граничными. Остальные грани будем называть внутренними. Для каждого вектора скорости  $\boldsymbol{\Omega}_{q,k}$  все граничные грани делятся на *входные*, где  $\boldsymbol{\Omega}_{q,k} \cdot \mathbf{n}(\mathbf{r}) < 0$ , и *выходные*, где  $\boldsymbol{\Omega}_{q,k} \cdot \mathbf{n}(\mathbf{r}) > 0$ . Определим следующие векторы:

$\Psi$  — содержит значения решения для всех векторов  $\boldsymbol{\Omega}_{q,k}$  в ячейках, на внутренних гранях и выходных граничных гранях;

${}^-\Psi$  — содержит значения решения на граничных входных гранях;

${}^+\Psi$  — содержит значения решения на граничных выходных гранях;

$\mathbf{f}$  — содержит значения граничного источника на входных гранях;

$\mathbf{Q}$  — содержит значения источника в ячейках;

$\Phi$  — содержит значения плотности скалярного потока в ячейках.

Тогда систему сеточных уравнений, аппроксимирующую задачу (1), (2), можно записать в форме

$$\hat{L}\Psi = \hat{S}\Psi + \mathbf{Q}, \quad {}^-\Psi = \mathbf{f}({}^+\Psi). \quad (3)$$

Матрицы  $\hat{L}$  и  $\hat{S}$  аппроксимируют дифференциальный и интегральный операторы в уравнении переноса (1).

Разобьем пространственную сетку на  $N$  непересекающихся подобластей. Тогда задача (3) принимает вид

$$\hat{L}_n \Psi_n = \hat{S}_n \Psi_n + \mathbf{Q}_n, \quad {}^-\Psi_n = \mathbf{f}({}^+\Psi_{n'}), \quad n' = 1, \dots, N, \quad n' \neq n. \quad (4)$$

Здесь  $n$  и  $n'$  — индексы подобластей.

При решении системы уравнений (4) будем использовать метод простой итерации. Уравнения метода простой итерации имеют вид

$$\hat{L}_n \Psi_n^{m+1} = \hat{S}_n \Psi_n^m + \mathbf{Q}_n, \quad {}^-\Psi_n^{m+1} = \mathbf{f}({}^+\Psi_{n'}^m), \quad n' = 1, \dots, N, \quad n' \neq n \quad (5)$$

или

$$\hat{L}_n \Psi_n^{m+1} = \hat{S}_n \Psi_n^m + \mathbf{Q}_n, \quad -\Psi_n^{m+1} = \mathbf{f}({}^+ \Psi_{n'}^{m+1}), \quad n' = 1, \dots, N, \quad n' \neq n, \quad (6)$$

где  $m$  — номер итерации. Равенства (5) и (6) отвечают ВJ- и BS-методам соответственно.

В BS-методе каждая простая итерация выполняется за несколько шагов; между шагами происходит обмен векторами  ${}^+ \Psi_{n'}$  между соседними MPI-процессами. Рассмотрим простой пример.

Пусть имеются две подобласти с прямоугольной сеткой. Тогда каждая итерация может быть выполнена за два шага (рис. 1). На каждом шаге расчет выполняется во всех ячейках для тех векторов скорости  $\Omega_{q,k}$ , для которых известны элементы векторов  ${}^- \Psi_n$ . Между шагами совершается обмен значениями решения на границе подобластей.

Если же в области задана неструктурированная треугольная сетка, то расчет для двух подобластей выполняется более чем за два шага (рис. 2). Например, для случая, представленного на рис. 2 слева, вектор  $\Omega$  направлен из ячейки 1 в ячейку 2, поэтому ячейка 2 не может быть рассчитана ранее ячейки 1. Аналогично, ячейка 4 рассчитывается после ячейки 3, а ячейка 6 — после ячейки 5. Поскольку в парах 1 и 2, 3 и 4, 5 и 6 ячейки относятся к разным подобластям, число шагов в расчете в итоге оказывается равным 4. Увеличение числа шагов означает увеличение количества MPI-обменов.

Из примеров, показанных на рис. 2, можно видеть, что на каждом шаге и для каждого вектора  $\Omega$  число рассчитываемых ячеек различно даже при равном числе ячеек в подобластях. Это приводит к простоям процессов.

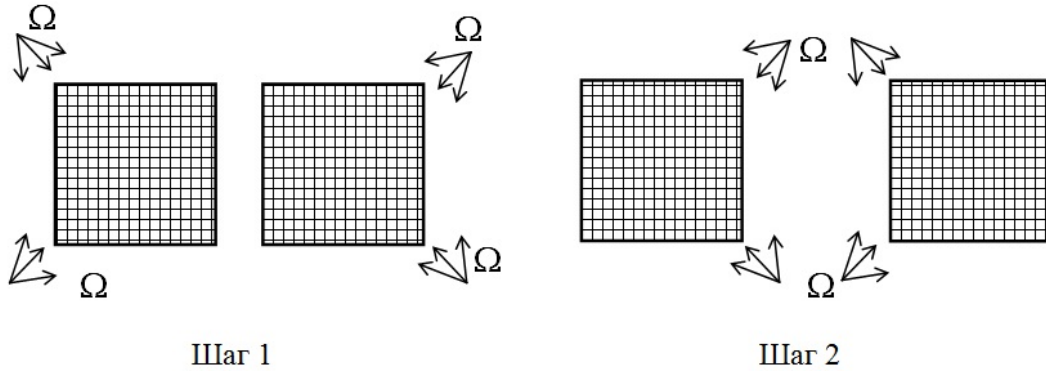


Рис. 1. Схема сквозного расчета области с прямоугольной сеткой (две подобласти)

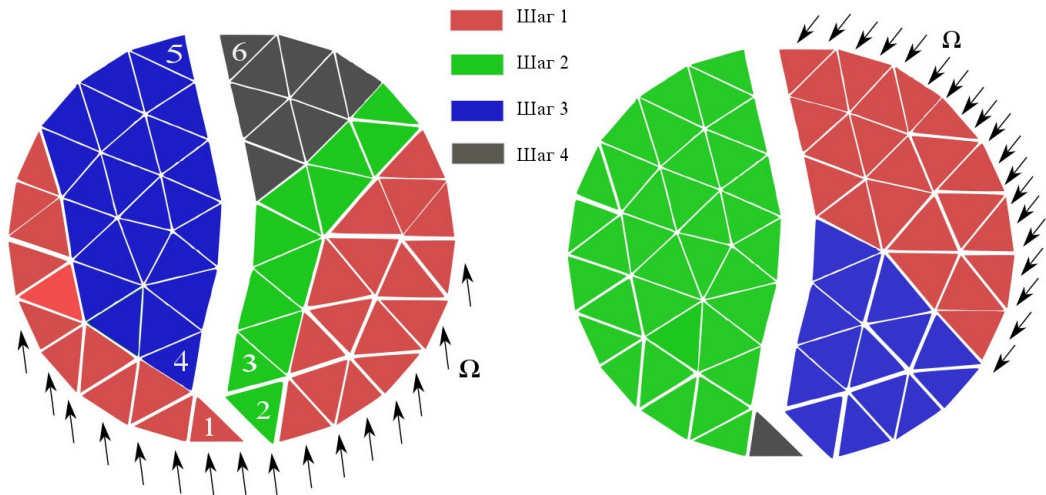


Рис. 2. Схема сквозного расчета области с треугольной сеткой для разных направлений потоков (две подобласти)

В КВА-методе сквозного счета для уменьшения простоев увеличивают число шагов, уменьшая порции рассчитываемых на каждом шаге ячеек и увеличивая число обменов. В представляемом здесь BS-методе, наоборот, с целью уменьшения числа обменов на каждом шаге рассчитывается максимально большая порция ячеек.

Для выравнивания числа ячеек, рассчитываемых на одном шаге всеми MPI-процессами, расчет на одном шаге выполняется не для одного направления  $\Omega$ , а для группы направлений. В качестве такой группы выбираются векторы  $\Omega$ , отвечающие одному квадранту. После расчета в одном квадранте можно начать выполнение асинхронных обменов значениями решения для данного квадранта. Можно надеяться, что обмены будут завершены к тому времени, когда снова начнется счет в этом квадранте, но уже на следующем шаге.

Общая для VJ- и BS-методов блок-схема расчета приведена на рис. 3. Основные этапы расчета — вычисления и MPI-обмены — выделены точечными рамками.

Отметим следующие особенности алгоритма:

1. В  $n$ -й подобласти для  $\ell$ -го шага и вектора  $\Omega_{q,k}$  расчет выполняется для набора ячеек  $C_{\ell,q,k,n}$ , причем  $\sum_{n=1}^N \sum_{\ell=1}^L |C_{\ell,q,k,n}| = J$ , где  $|C_{\ell,q,k,n}|$  — число ячеек в наборе  $C_{\ell,q,k,n}$ . Для VJ-метода число шагов  $L = 1$ , для BS-метода всегда  $L > 1$ .
2. Составление наборов ячеек  $C_{\ell,q,k,n}$  и определение числа шагов  $L$  выполняется один раз, до начала итерационного процесса (не динамически в процессе счета).
3. Между асинхронными пересылками выполняются вычисления для всех векторов скорости из одного квадранта. Это дает возможность уменьшить дисбаланс числа ячеек, рассчитываемых на одном шаге на разных MPI-процессах, и занять MPI-процессы вычислениями в других квадрантах на время выполнения пересылок. Дисбаланс на одном шаге характеризуется долей рассчитываемых "лишних" ячеек

$$D_{\ell} = \left( \max_n \sum_{q=1}^4 \sum_{k=1}^K |C_{\ell,q,k,n}| - \min_n \sum_{q=1}^4 \sum_{k=1}^K |C_{\ell,q,k,n}| \right) / (JK).$$

Общий дисбаланс находится по формуле

$$D = \sum_{\ell=1}^L D_{\ell}. \quad (7)$$

Отметим, что в VJ-методе при одинаковом числе ячеек в каждой подобласти все величины  $D_{\ell} = 0$ , т. е.  $D = 0$ .

4. MPI-обмены выполняются с буферизацией. Буферизация помогает сократить время инициализации пересылки данных.
5. Условие отражения от внешних границ в обоих методах в настоящий момент реализовано итерационно.

## 2. Численные результаты

Тестирование рассматриваемых параллельных алгоритмов выполняется на тестовой задаче DogLeg [22] о глубоком проникновении излучения от локализованного источника через неоднородную среду (рис. 4). Здесь область расчета заполнена плотной средой (сечение  $\sigma = 0,1$ ), содержащей прозрачный канал (сечение  $\sigma = 0,0001$ ). Сечение рассеяния в обоих материалах  $\sigma_s = \sigma/2$  и не зависит от  $\Omega \cdot \Omega'$ . На левой ( $x = 0$ ), передней ( $y = 0$ ) и нижней ( $z = 0$ ) границах задано условие отражения.

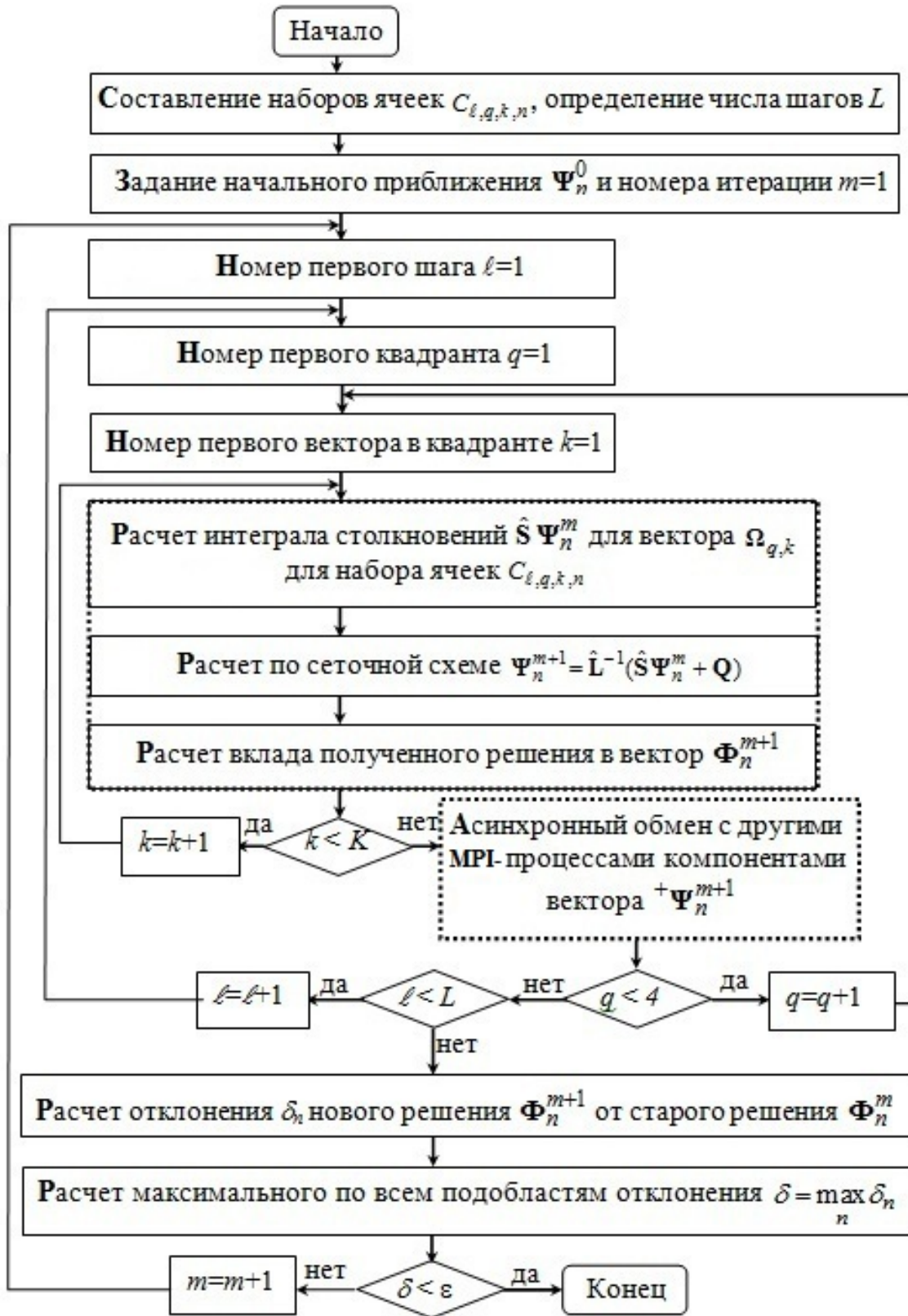


Рис. 3. Блок-схема программы "Радуга Т" (расчет для нейтронов с одинаковой энергией)

При тестировании используется квадратура Карлсона  $ES_{12}$ , в каждом квадранте содержащая  $K = 42$  вектора. Пространственная сетка из тетраэдров строится с помощью платформы Salome [23]. Далее полученная сетка сгущается следующим образом: наиболее крупные ячейки последовательно разбиваются на две до тех пор, пока число ячеек не увеличится вдвое. Таким образом, получается последовательность сеток

$$A : J = 18\,005; \quad B : J = 36\,004; \quad C : J = 72\,003; \quad D : J = 144\,001; \quad E : J = 288\,005. \quad (8)$$

Для решения задачи (1), (2) используется метод конечных элементов [24].

Расчеты выполняются на гибридном кластере К-100 [25] общего пользования с распределенной памятью.

Задаются 2, 4, 8, 16 и 32 подобласти. Для разбиения на подобласти используется включенный в программу "Радуга Т" координатный метод [26]. Пример разбиения на четыре подобласти для сетки А показан на рис. 5. Отметим, что координатный метод порождает КВА-подобное разбиение на подобласти, как в [19]. Результаты тестирования алгоритмов распараллеливания на сетках А, В, С, D, Е приведены на рис. 6–21.

Можно видеть, что число шагов в BS-методе достаточно велико (см. рис. 6) и в целом растет с ростом числа подобластей  $N$  и числа ячеек  $J$ . В ВJ-методе с ростом  $N$  растет число итераций.

Будем полагать, что время обменов разделяется на время организации и инициализации пересылок данных и время ожидания получения от соседнего MPI-процесса необходимых для расчета компонент вектора  $-\Psi_n^{m+1}$ . Таким образом, время расчета состоит из времен вычислений, организации пересылок, инициализации пересылок и ожидания данных.

Доля времени инициализации пересылок пренебрежимо мала (см. рис. 9).

Организация пересылок (включает в себя сбор и перенос в буфер пересылаемых данных) в BS-методе занимает много больше времени, чем в ВJ-методе (см. рис. 12). Это происходит в основном из-за излишнего объема пересылаемых данных. Время организации пересылок в BS-методе может быть сокращено за счет модернизации алгоритма. Однако доля этого этапа в целом не превышает 0,12 (см. рис. 8).

Дополнительные траты времени в основном связаны с ожиданием получения данных от других процессов (см. рис. 10). Можно предположить, что ожидание возникает из-за дисбаланса числа

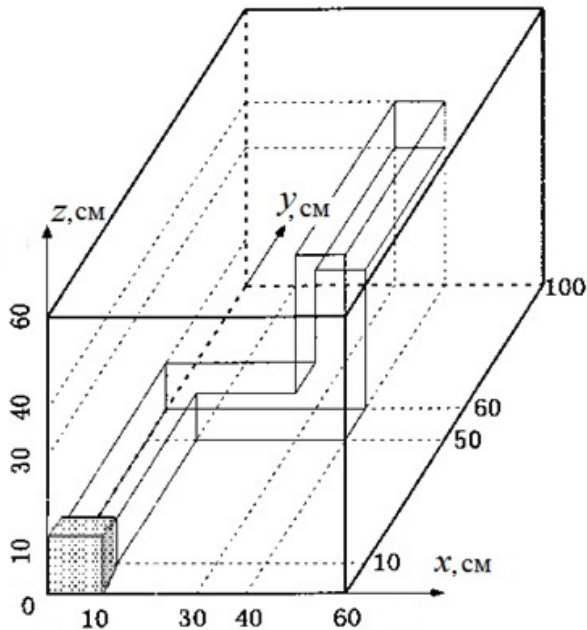


Рис. 4. Область расчета тестовой задачи DogLeg

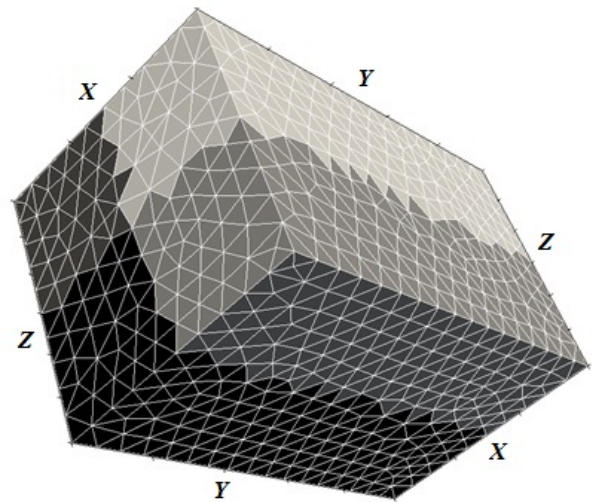


Рис. 5. Разбиение сетки А на четыре подобласти

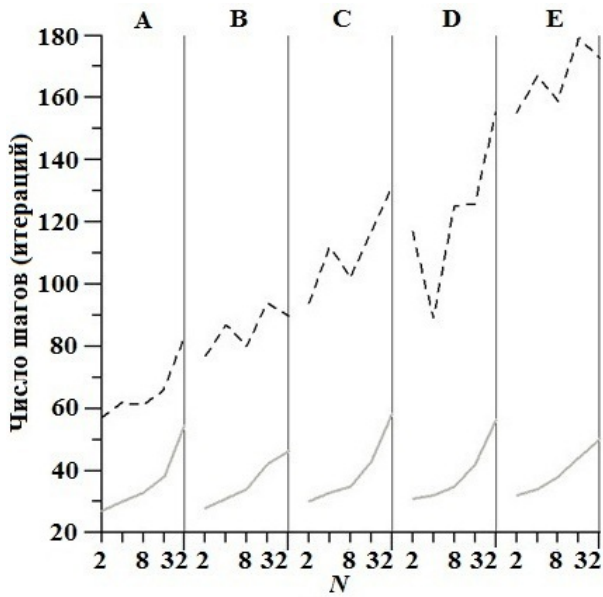


Рис. 6. Число шагов на одной итерации в BS-методе (---) и число итераций в VJ-методе (—)

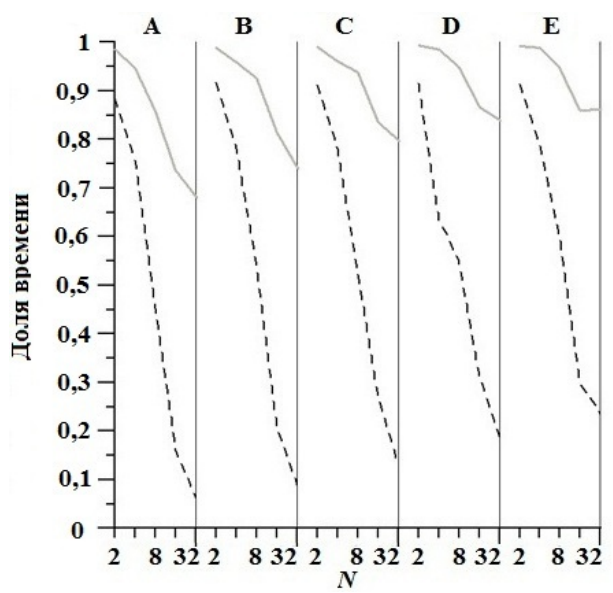


Рис. 7. Доля времени вычислений: --- — BS-метод; — — VJ-метод

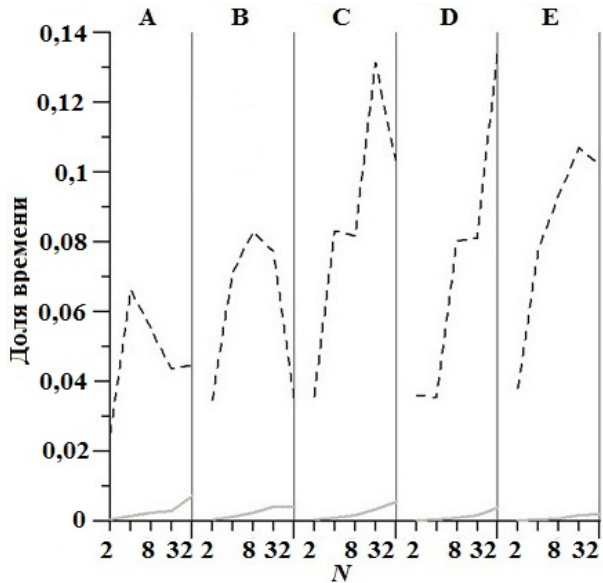


Рис. 8. Доля времени организации пересылок: --- — BS-метод; — — VJ-метод

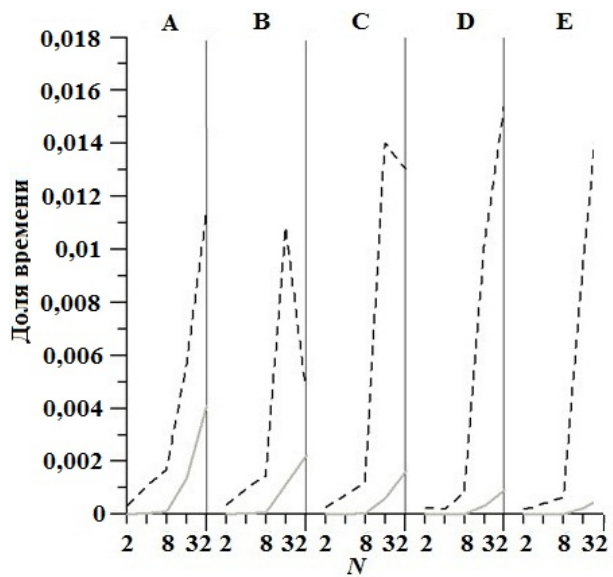


Рис. 9. Доля времени инициализации пересылок: --- — BS-метод; — — VJ-метод

ячеек, рассчитываемых в разных подобластях. Однако зависимость времени ожидания от числа подобластей  $N$  не соответствует зависимости от  $N$  дисбаланса  $D$  (см. рис. 13, и рис. 18). С другой стороны, время ожидания существенно и при отсутствии дисбаланса, т. е. в VJ-методе (см. рис. 10). Можно заключить, что ожидание возникает из-за задержек при прохождении данных через коммуникационную сеть компьютера общего пользования. Подчеркнем, что на одном шаге время ожидания в BS-методе много меньше, чем в VJ-методе, а на состоящей из многих шагов одной итерации — наоборот (см. рис. 13, 14). Это показывает, что пересылки, действительно, выполняются асинхронно, но поток данных замедляется при прохождении через коммуникационную сеть.



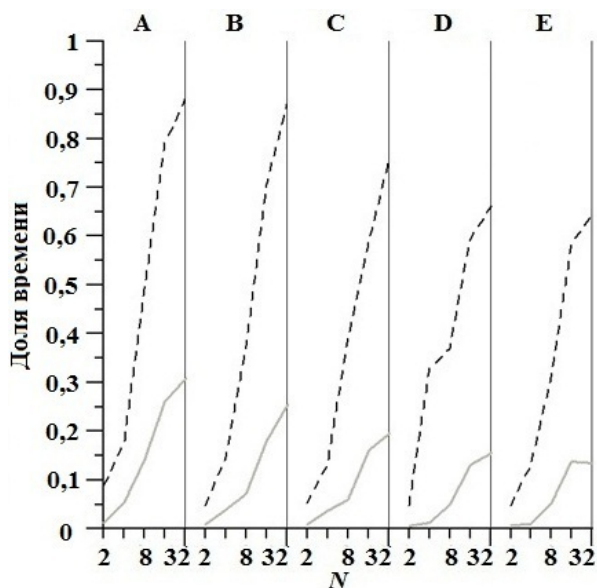


Рис. 10. Доля времени ожидания: - - - - BS-метод; — — — VJ-метод

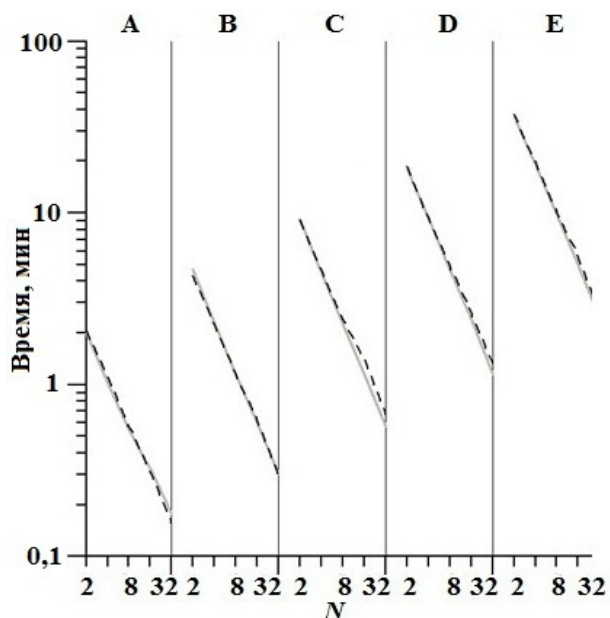


Рис. 11. Время вычислений на одной итерации: - - - - BS-метод; — — — VJ-метод

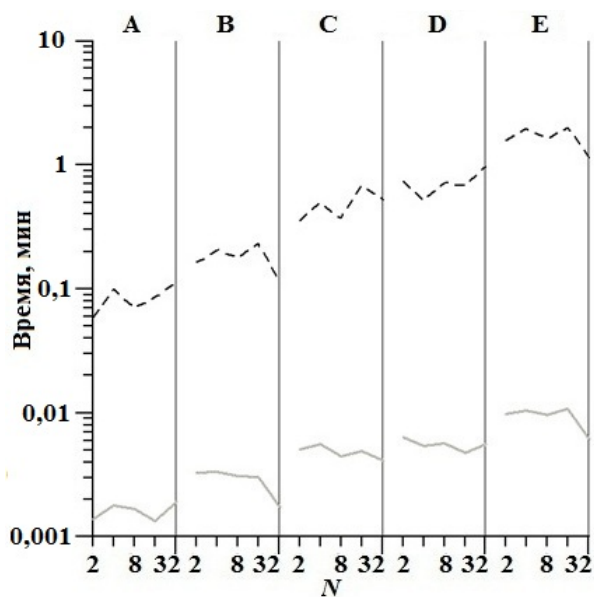


Рис. 12. Время организации пересылок на одной итерации: - - - - BS-метод; — — — VJ-метод

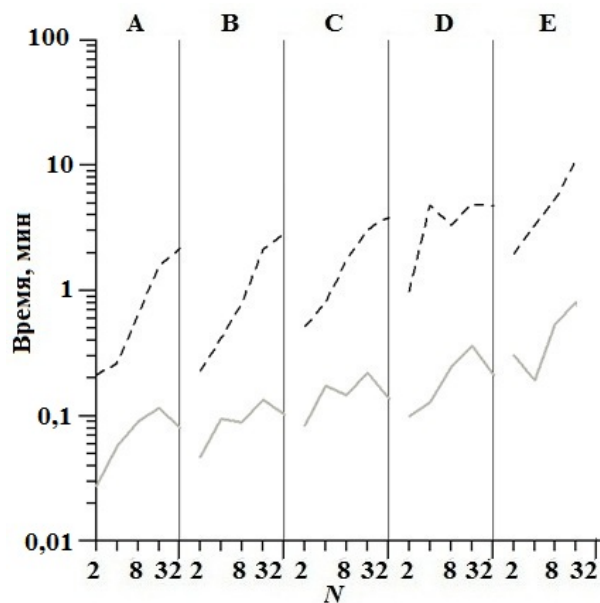


Рис. 13. Время ожидания на одной итерации: - - - - BS-метод; — — — VJ-метод

Время вычислений на одной итерации обоими методами сравнимо (см. рис. 11), хотя в VJ-методе выполняется один шаг, а в BS-методе — много шагов (см. рис. 6).

Однако из-за дополнительных временных затрат на MPI-обмены, главным образом ожидание данных, время расчета одной итерации в BS-методе всегда больше, чем в VJ-методе (см. рис. 15). Хотя число итераций в VJ-методе растет с числом подобластей (см. рис. 6), а в BS-методе число итераций от числа подобластей не зависит, время VJ-метода в рассматриваемой задаче оказывается больше, чем время BS-метода, лишь если число ячеек в одной подобласти достаточно велико (см. рис. 16, 17).

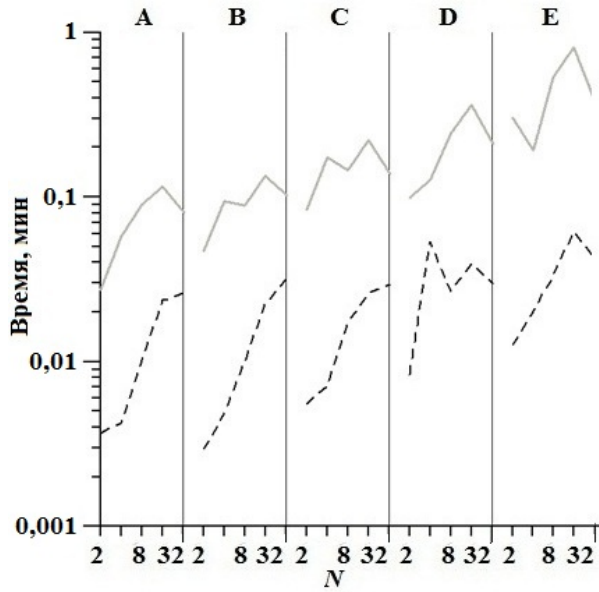


Рис. 14. Время ожидания на одном шаге: - - - - BS-метод; — — VJ-метод

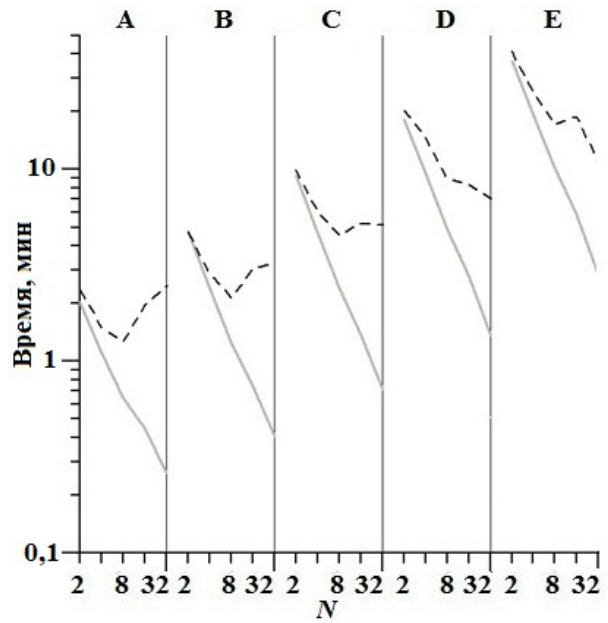


Рис. 15. Время расчета одной итерации: - - - - BS-метод; — — VJ-метод

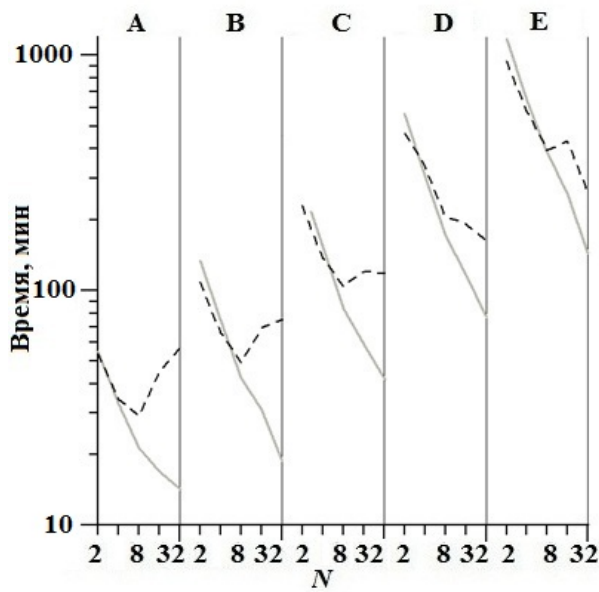


Рис. 16. Время расчета всех итераций: - - - - BS-метод; — — VJ-метод

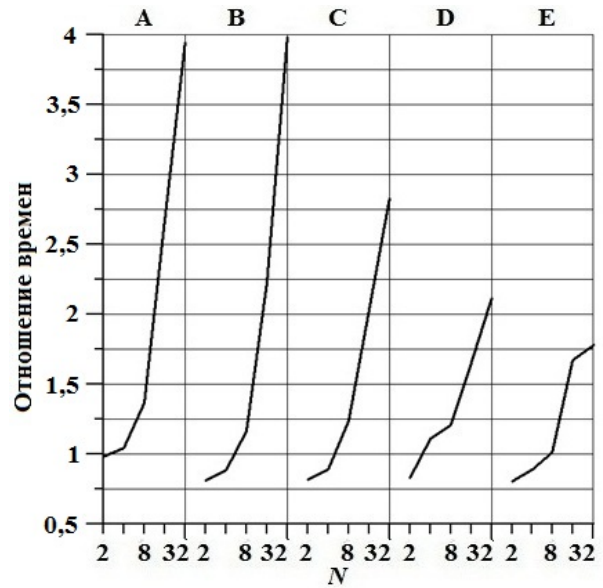


Рис. 17. Отношение времени расчета всех итераций BS-методом ко времени расчета VJ-методом

При увеличении числа ячеек сетки вдвое (см. (8)) число шагов увеличивается приблизительно на 25–30% (см. рис. 6). Соответственно при росте числа ячеек доля времени ожиданий уменьшается (см. рис. 10). Поэтому уменьшается и отношение времен счета BS/VJ (см. рис. 17). Для достаточно большого числа ячеек сетки временные затраты BS-метода меньше временных затрат VJ-метода.

На рис. 19 показаны эффективности распараллеливания вычислений на одной итерации, найденные по формуле

$$Eff(N) = \frac{2T(2)I(N)}{I(2)NT(N)},$$

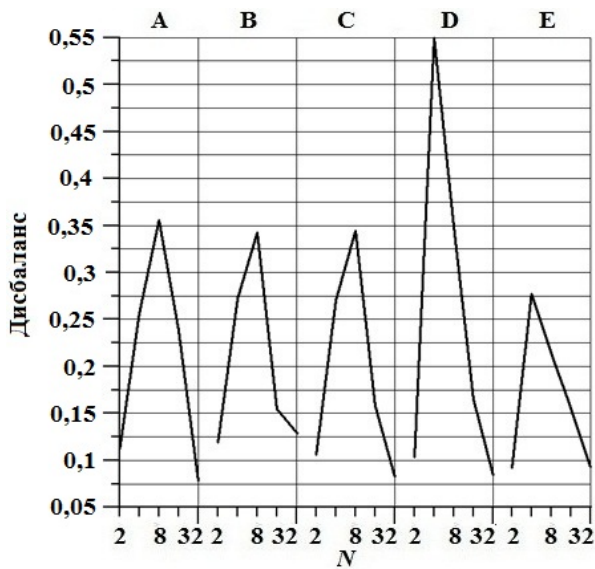


Рис. 18. Дисбаланс  $D$  числа ячеек

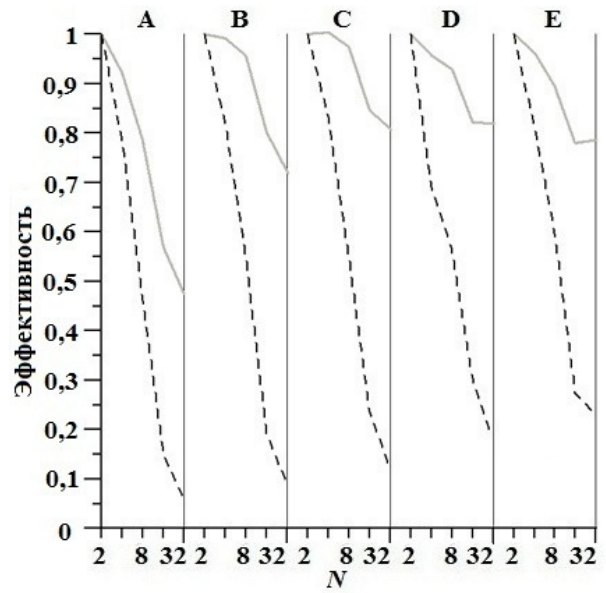


Рис. 19. Эффективность распараллеливания на одной итерации: --- — BS-метод; — — — VJ-метод

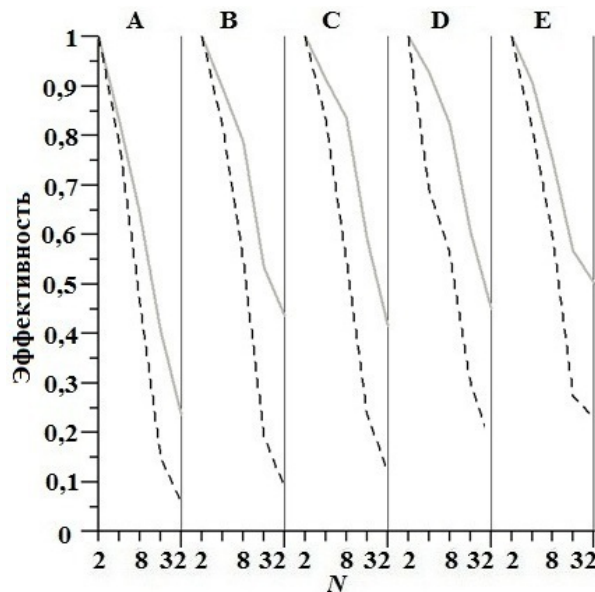


Рис. 20. Эффективность распараллеливания всех итераций: --- — BS-метод; — — — VJ-метод

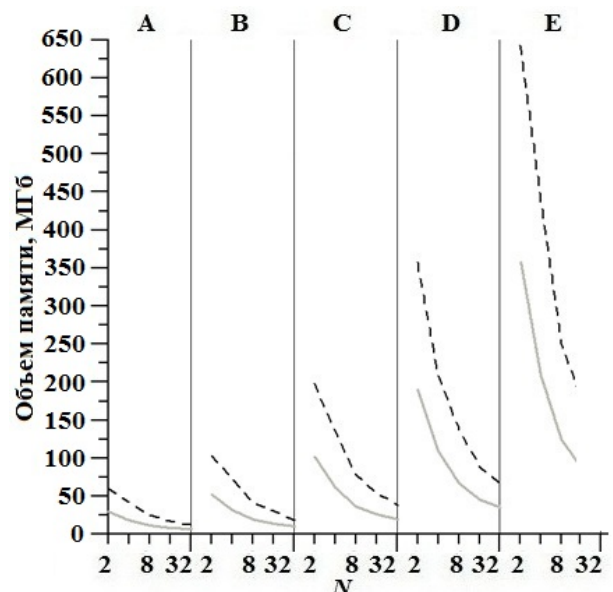


Рис. 21. Объем памяти на одну подобласть: --- — BS-метод; — — — VJ-метод

где  $T(N)$ ,  $I(N)$  — время расчета и число итераций при использовании  $N$  подобластей. На рис. 20 — эффективности полных расчетов, найденные по формуле

$$Eff'(N) = \frac{2T(2)}{NT(N)}.$$

Можно видеть, что эффективности VJ-метода много выше эффективностей BS-метода, причем тем выше, чем больше число подобластей, хотя при увеличении числа ячеек сетки обе эффективности обоих методов растут. Однако выигрыш в эффективности VJ-метода по сравнению с BS-методом для полного расчета много меньше, чем для одной итерации. Отметим, что обычно эффективность распараллеливания вычислений при решении уравнения переноса вычисляется для одной итерации.

Наконец, на рис. 21 приведены объемы оперативной памяти, необходимые для расчета каждым методом. В BS-методе требуется дополнительно хранить значения решения на тех внутренних гранях, которые являются общими для ячеек, рассчитываемых на разных шагах. Число таких граней значительно, поскольку велико число шагов (см. рис. 6), а хранить значения решения в BS-алгоритме (см. рис. 3) необходимо для всех векторов  $\Omega_{q,k}$ . В результате в рассматриваемой модельной задаче для BS-метода требуется примерно в два раза больше оперативной памяти для одной подобласти, чем для VJ-метода.

Также следует отметить, что объем требуемой памяти для обоих методов возрастает с увеличением числа ячеек и уменьшается с ростом числа подобластей  $N$ .

## Заключение

Рассматриваются два метода распараллеливания вычислений, используемые в программе "Радуга Т" для решения уравнения переноса нейтронов на неструктурированных сетках. В методе VJ простые итерации выполняются как по интегралу столкновений, так и по значениям решения на границах подобластей. В методе BS, как и в известном КВА-подходе, на каждой простой итерации выполняется сквозной счет во всей области. Сквозной счет в BS-методе выполняется за несколько шагов, перемежаемых MPI-обменами. В VJ-методе одна итерация выполняется за один шаг.

Отличие BS-метода от оригинального КВА-метода состоит в том, что BS-метод нацелен на сокращение числа MPI-обменов за счет некоторого дисбаланса числа рассчитываемых на каждом шаге в каждой подобласти ячеек, тогда как в КВА-методе баланс числа ячеек достигается за счет увеличения числа шагов.

Проведенное тестирование BS- и VJ-методов на модельной задаче показало, что для BS-метода выполнение одной простой итерации требует больших временных затрат, чем для VJ-метода. Дополнительное время в BS-методе главным образом тратится на ожидание получения данных от соседних процессов. Ожидание возникает за счет задержек при прохождении большого числа порций данных через коммуникационную сеть компьютера общего пользования.

BS-метод требует меньше времени, чем VJ-метод, только если число ячеек сетки в одной подобласти достаточно большое.

Эффективность VJ-метода на одной итерации в рассматриваемой задаче при увеличении числа подобластей до 32 не снижается ниже 0,5, тогда как в BS-методе падает до 0,1. Однако эффективность VJ-метода в полном расчете (все итерации) при увеличении числа подобластей до 32 может снижаться до 0,2.

BS-метод требует заметно больших объемов оперативной памяти, чем VJ-метод.

Основные дополнительные затраты времени в BS-методе связаны с ожиданием данных от других процессов. Для сокращения этих затрат необходимо либо ускорить прохождение данных по коммуникационной сети, либо уменьшить число обменов, т. е. число шагов на одной простой итерации. Сокращение числа шагов возможно при *выглаживании* границ подобластей. Однако такое выглаживание приводит к ухудшению качества сетки.

Отметим, что значительное время ожидания может быть связано с тем, что тестирование выполняется в реальных условиях, когда на многопроцессорном кластере присутствуют задачи и других пользователей. В теоретической оценке эффективности предполагается, что MPI-обмены выполняются мгновенно. Однако при оценке эффективности распараллеливания в реальных условиях затраты времени на ожидание данных необходимо учитывать.

Также время счета в BS-методе может быть несколько сокращено модернизацией модуля организации пересылок данных.

Здесь рассмотрены методы распараллеливания простых итераций при решении уравнения переноса. В задачах о протяженных и/или сильно рассеивающих средах чаще используются двухшаговые итерационные алгоритмы, включающие расчет и прибавление к решению, полученному после простой итерации, ускоряющей сходимость поправки. Вопрос об эффективности распараллеливания двухшаговых алгоритмов нуждается в отдельном исследовании.

## Список литературы

1. *Моряков А. В., Исаков Н. Ш., Спиридонов А. А.* Решение задач переноса нейтронного и гамма излучения с использованием многопроцессорных вычислительных систем // Суперкомпьютерные дни в России. Тр. межд. конф. М.: МГУ, 2017. С. 724–732.  
*Moryakov A. V., Isakov N. Sh., Spiridonov A. A.* Resheniye zadach perenosa neytronnogo i gamma izlucheniya s ispolzovaniem mnogoprotsesornykh vychislitelnykh system // Superkompyuternye dni v Rossii. Tr. Mezhd. konf. M.: MGU, 2017. S. 724–732.
2. *Lenain R., Masiello E., Damian F., Sanchez R.* Domain decomposition method for 2D and 3D transport calculations using hybrid MPI/OPENMP parallelizm // Proc. Joint Int. Conf. on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method. Nashville, Tennessee, USA. April 19–23, 2015. LaGrange Park, IL: American Nuclear Society, 2015. <https://hal-cea.archives-ouvertes.fr/cea-02506817/document>.
3. *Yessayan R., Azmy Y., Schunert S.* Development of a parallel performance model for the THOR neutral particle transport code // Proc. Int. Conf. on Mathematics & Computational Methods Applied to Nuclear Science & Engineering. Jeju, Korea. April 16–20, 2017. <https://www.osti.gov/servlets/purl/1369430>.
4. *Chen Y., Zhang B., Zhang L., Zheng J., Zheng Y., Liu C.* ARES: a parallel discrete ordinates transport code for radiation shielding applications and reactor physics analysis // Hindawi Science and Technology of Nuclear Installations. 2017. Article ID 2596727.
5. *Adams M. P., Adams M. L., Hawkins W. D., Smith T., Rauchwerger L., Amato N. M., Bailey T. S., Falgout R. D.* Provably optimal parallel transport sweeps on regular grids // Proc. Int. Conf. on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M & C 2013). Sun Valley, Idaho, USA. May 5–9, 2013. LaGrange Park, IL: American Nuclear Society, 2013. <https://experts.illinois.edu/en/publications/provably-optimal-parallel-transport-sweeps-on-regular-grids>.
6. *Pautz Sh. D., Bailey T. S.* Parallel deterministic transport sweeps of structured and unstructured meshes with overloaded mesh decompositions // Nuclear Science and Engineering. 2017. Vol. 185. P. 70–77. DOI: 10.13182/NSE16-34.
7. *Gong C., Liu J., Chi L., Huang H., Fang J., Gong Z.* GPU accelerated simulations of 3D deterministic particle transport using discrete ordinates method // J. Comp. Phys. 2011. Vol. 230. P. 6010–6022. DOI:10.1016/j.jcp.2011.04.010.
8. *Deakin T., McIntosh-Smith S., Gaudin W.* Many-core acceleration of a discrete ordinates. Transport mini-app at extreme scale // Proc. Int. Conf. on High Performance Computing. Frankfurt, Germany. June 19–23, 2016. P. 429–448. DOI: 10.1007/978-3-319-41321-1\_22. [https://www.researchgate.net/deref/http%3A%2F%2Fdx.doi.org%2F10.1007%2F978-3-319-41321-1\\_22](https://www.researchgate.net/deref/http%3A%2F%2Fdx.doi.org%2F10.1007%2F978-3-319-41321-1_22).
9. *Алексеев А. В., Беляков И. М., Бочков А. И., Евдокимов В. В., Ириничев Е. Ф., Морозов В. Ю., Москвин А. Н., Нуждин А. А., Пепеляев М. П., Резчиков В. Ю., Сучкова В. В., Шагалиев Р. М., Шарифуллин Э. Ш., Шемякина Т. В., Шумилин В. А.* Методика САТУРН-2005. Математические модели. Алгоритмы и программы решения многомерных задач переноса частиц и энергии // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2013. Вып. 4. С. 17–30.  
*Alekseev A. V., Belyakov I. M., Bochkov A. I., Evdokimov V. V., Irinichev E. F., Morozov V. Yu., Moskvina A. N., Nuzhdin A. A., Pepelyaev M. P., Rezchikov V. Yu., Suchkova V. V., Shagaliev R. M., Sharifullin E. Sh., Shemyakina T. V., Shumilin V. A.* Metodika SATURN-2005. Matematicheskie modeli. Algoritmy i programmy resheniya mnogomernykh zadach perenosa chastits i energii // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsesov. 2013. Vyp. 4. S. 17–30.
10. *Baker R. S.* An SN algorithm for modern architectures // Nuclear Science and Engineering. 2017. Vol. 185. P. 107–116. DOI: 10.13182/NSE15-124.

11. Бочков А. И., Нuzhдин А. А. Параллельный алгоритм решения трехмерного кинетического уравнения переноса. Программа ПАУК для тестирования многопроцессорных вычислительных систем // Параллельные вычислительные технологии (ПАВТ'2008): Тр. межд. науч. конф. (С.Петербург, 28 января — 1 февраля 2008 г.). Челябинск: ЮУрГУ, 2008. С. 39—47.  
*Bochkov A. I., Nuzhdin A. A. Parallelny algoritm resheniya kineticheskogo uravneniya perenosa. Programma PAUK dlya testirovaniya mnogoprotsesornykh vychislitelnykh system // Parallelnye vychislitelnye tekhnologii (PaVT'2008): Tr. mezhd. nauch. konf. (S.Peterburg, 28 yanvaryaya — 1 fevralya 2008 g.). Chelyabinsk: YuUrGU, 2008. S. 39—47.*
12. Волощенко А. М. Адаптивные положительные аппроксимации и согласованная КР1 схема ускорения итераций для уравнения переноса в задачах радиационной защиты: Автореф. дис. доктора физ.-мат. наук. М.: ИПМ им. М. В. Келдыша РАН, 2015.  
*Voloshchenko A. M. Adaptivnyye polozhitelnye approksimatsii i soglasovannaya KR1 skhema uskoreniya iteratsiy dlya uravneniya perenosa v zadachakh radiatsionnoy zashchity: Avtoref. dis. doktora fiz.-mat. nauk. M.: IPM im. M. V. Keldysha RAN, 2015.*
13. Cheng T., Mo Z., Zhang G., Yan J., Xu Q., Fu Y., Deng L. Radiation shielding calculation using the capabilities of large-scale mesh generation and efficient parallel computing in JSNT on tens of thousands of processors // Proc. Int. Conf. on Mathematics & Computational Methods Applied to Nuclear Science & Engineering. Jeju, Korea. April 16—20, 2017. <https://www.semanticscholar.org/paper/Radiation-Shielding-Calculation-using-the-of-Mesh-Cheng-Mo/aa89e64b78eb45581170216c422fdce7b2c377c0#paper-header>.
14. Jarrell J. J., Evans T. M., Godfrey A. T. Full core reactor analysis: running Denovo on Jaguar // Nucl. Sci. Eng. 2013. Vol. 175, No 3. P. 283—291. DOI: 10.13182/NSE12-60.
15. Байдин Д. Ф., Аристова Е. Н. Параллельный код QUDIFF для расчета критических параметров реактора на быстрых нейтронах в трехмерной гексагональной геометрии // Математическое моделирование. 2016. Т. 28, № 1. С. 107—116.  
*Baydin D. F., Aristova E. N. Parallelny kod QUDIFF dlya raschyeta kriticheskikh parametrov reaktora na bystrykh neytronakh v tryekhmernoy geksgonalnoy geometrii // Matematicheskoe modelirovanie. 2016. T. 28, № 1. S. 107—116.*
16. Ghaddar T. H., Ragusa J. C. An approach for load balancing massively parallel transport sweeps on unstructured grids // Proc. Int. Conf. on Mathematics & Computational Methods Applied to Nuclear Science & Engineering. Jeju, Korea. April 16—20, 2017. <https://pdfs.semanticscholar.org/13d5/7ba48b421c4e301af981fcb6bce525115d9e.pdf>.
17. Vega R. M., Adams M. L. Transport sweeps using an improved slice balance approach with LDFE and GPU acceleration // Proc. Int. Conf. on Mathematics & Computational Methods Applied to Nuclear Science & Engineering. Jeju, Korea, April 16—20, 2017. [https://www.kns.org/files/int\\_paper/paper/MC2017\\_2017\\_1/P056S01-11VegaR.pdf](https://www.kns.org/files/int_paper/paper/MC2017_2017_1/P056S01-11VegaR.pdf).
18. Pautz Sh. D. An algorithm for parallel  $S_n$  sweeps on unstructured meshes // Nucl. Sci. and Eng. 2002. Vol. 140, No 2. P. 111—136. <https://doi.org/10.13182/NSE02-1>.
19. Plimpton S. J., Hendrickson B., Burns Sh. P., McLendon W. III, Rauchwerger L. Parallel  $S_n$  sweeps on unstructured grids: algorithms for prioritization, grid partitioning, and cycle detection // Nucl. Sci. and Eng. 2005. Vol. 150. P. 267—283. DOI:10.13182/NSE150-267.
20. Colomer G., Borrell R., Trias F. X., Rodriguez I. Parallel algorithms for  $S_n$  transport sweeps on unstructured meshes // J. Comp. Phys. 2013. Vol. 232. P. 118—135. DOI:10.1016/j.jcp.
21. Николаева О. В., Басс Л. П., Гайфуллин С. А. Параллельные алгоритмы решения уравнения переноса нейтронов в защитах ядерных реакторов на неструктурированных тетраэдрических сетках в программе "Радуга Т" // Параллельные вычислительные технологии (ПАВТ'2018): Тр. межд. науч. конф. (2—6 апреля 2018 г., г. Ростов-на-Дону). Челябинск: ЮУрГУ, 2018. С. 319—331.  
*Nikolaeva O. V., Bass L. P., Gayfulin S. A. Parallelnye algoritmy resheniya uravneniya perenosa*

- neutronov v zashchitakh yadernykh reaktorov na nestrukturirovannykh tetraedricheskikh setkakh v programme "Raduga T" // Parallelnye vychislitelnye tekhnologii (PaVT'2018): Tr. mezhd. nauch. konf. (2–6 aprelya 2018 g., g. Rostov-na-Donu). Chelyabinsk: YuUrGU, 2018. S. 319–331.
22. *Takeda T., Ikeda H.* 3-D Neutron transport benchmarks // J. Nucl. Sci. Tech. 1991. Vol. 28, No 7. P. 656–669. DOI:10.3327/jnst.28.656.
23. The Open Source Integration Platform for Numerical Simulation. <http://www.salome-platform.org/>.
24. *Николаева О. В., Казанцева А. С.* Сравнение свойств сеточных схем для решения уравнения переноса на неструктурированных тетраэдрических сетках // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2019. Вып 1. С. 3–18.  
*Nikolaeva O. V., Kazantseva A. S.* Sravneniye svoystv setochnykh skhem dlya resheniya uravneniya perenosa na nestrukturirovannykh tetraedricheskikh setkakh // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsesov. 2019. Vyp 1. S. 3–18.
25. Гибридный вычислительный кластер К-100. <http://www.kiam.ru/MVS/resourse/k100.html>.  
Gibridny vychislitelny klaster K-100. <http://www.kiam.ru/MVS/resourse/k100.html>.
26. *Николаева О. В., Гайфулин С. А., Басс Л. П.* О декомпозиции неструктурированной сетки при решении уравнения переноса нейтронов на параллельных компьютерах // Параллельные вычислительные технологии (ПаВТ'2019): Тр. межд. науч. конф. (2–4 апреля 2019 г., г. Калининград). Челябинск: ЮУрГУ, 2019. С. 362–372.  
*Nikolaeva O. V., Gayfulin S. A., Bass L. P.* O dekompozitsii nestrukturirovannoy setki pri reshenii uravneniya perenosa neytronov na parallelnykh kompyuterakh // Parallelnye vychislitelnye tekhnologii (PaVT'2019): Tr. mezhd. nauch. konf. (2–4 aprelya 2019 g., g. Kaliningrad). Chelyabinsk: YuUrGU, 2019. S. 362–372.

Статья поступила в редакцию 24.03.20.

COMPUTATION PARALLELING METHODS IN THE "RADUGA T" CODE TO SOLVE THE NEUTRON TRANSPORT EQUATION ON UNSTRUCTURED GRIDS / O. V. Nikolaeva\* (M. V. Keldysh IAM of RAS, Moscow).

The problem of paralleling computations when solving the integro-differential transport equation using unstructured grids is considered. Paralleling is performed by fragmenting a spatial grid into subdomains, with computations being carried out on its own MPI-process for each subdomain. Two computation paralleling methods implemented in the "Raduga T" code are discussed: the BJ (Block-Jacobi) and BS (Block-Seidel) methods. In the BJ method simple iterations with respect to the collision integral and solution values obtained at the boundaries of spatial domains are performed. In the BS method, which follows the idea of the well-known KBA (Koch-Baker-Alcouffe) method, simple iterations are performed with respect to the collision integral alone. Hence, the BS method has a less number of simple iterations, than the BJ method, however, a longer time is spent to perform them. There has been examined the efficiency of the BJ and BS methods of paralleling computations in the solution of a model problem using the "Raduga T" code.

*Keywords:* the neutron transport equation, unstructured grids, parallel algorithms, the Block-Jacobi method, the Block-Seidel method, the KBA method.

---

\*Nikolaeva Olga Vasilyevna, senior scientist  
*e-mail:* nika@kiam.ru