когда к результатам работы одного алгоритма можно применить другой алгоритм фильтрации, в качестве дополнительного параметра в функцию createFilter передается либо имя элемента сцены, к которому необходимо применить новый алгоритм фильтрации, либо непосредственно объект ранее созданного фильтра.

Далее в качестве примера макропрограммы приведено задание параметров алгоритма фильтрации «Интервал».

```
filter = createFilter(INTERVAL)
filter.setParams(«Material», 0, 5)
filter.build()
```

В данном примере функция *setParams* устанавливает параметры фильтра, указывая, что в результат алгоритма должны попасть ячейки, значение сеточной величины с именем *«Material»* в диапазоне от 0 до 5.

В рамках работы, описанной в докладе, обеспечена возможность обработки данных в параллельной системе постобработки ScientificView, не только используя графический интерфейс пользователя системы, но и посредством команд разработанного макроязыка. При этом использование команд макроязыка зачастую позволяет автоматизировать процесс обработки данных и существенно сэкономить время пользователя.

Литература

1. Потехин А. Л., Логинов И. В., Козачек Ю. В. и др. ScientificView – параллельная система постобработки результатов, полученных при численном моделировании физических процессов // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2008. Вып. 4. С. 37–45.

2. Потехин А. Л., Логинов И. В., Тарасов В. И. и др. ScientificView – параллельная система постобработки результатов, полученных при численном моделировании физических процессов // XVIII Международная конференция по компьютерной графике и зрению «Графикон»: сб. трудов. Москва, 23–27 июня 2008. С. 192–198.

РАЗВИТИЕ ПРОГРАММЫ *LUCKY* ДЛЯ РЕШЕНИЯ УСЛОВНО-КРИТИЧЕСКИХ И НЕСТАЦИОНАРНЫХ ЗАДАЧ НЕЙТРОННОЙ ФИЗИКИ

А. В. Моряков

Национальный исследовательский центр «Курчатовский институт», г. Москва

Введение

Основными элементами современной вычислительной техники являются суперкомпьютеры – вычислительные машины с большим числом вычислительных ядер, способных работать в параллельном режиме и использовать для решаемой задачи огромные ресурсы всей системы. Суперкомпьютеры позволяют применять максимально детализированные модели для решаемых задач, использовать пространственную сетку большой размерности и получать результат за небольшое расчетное время.

Задача на собственное значение (нахождение $K_{\text{эфф}}$) является одной из основных задач нейтронной физики активной зоны реактора. Повышенные требования к точности решения этой задачи определяются критериями безопасности в работе ЯЭУ (ядерно-энергетических установок).

Программа *LUCKY-A* обладает широкими возможностями. Она создана в результате развития программы *LUCKY* [1–5] и применяется для решения задач защиты ЯЭУ от нейтронного и гамма излучения, задач на собственное значение (модуль *LUCKY*), а также решения широкого круга нестационарных задач, описываемых уравнением переноса, задач с делением и распределенным источником, зависящим от времени (модуль *LUCKY_TD*). Для моделирования пространственной кинетики реакторов предусмотрен расчет задач с последовательным внесением реактивности по заданному сценарию. В программе реализовано решение условно-критической задачи, получаемое как решение стационарного уравнения переноса, так и нестационарного уравнения переноса (через динамический процесс). Кроме того, в *LUCKY-A* возможен расчет нестационарных полей температур с заданными или рассчитанными источниками удельного тепловыделения (модуль *LUCKY_HEAT*).

Особенностью программы *LUCKY-A* является использование *XYZ* геометрии, при малых пространственных шагах по пространственным осям эта геометрия может быть применена как универсальная. Программа разработана для применения на вычислительных машинах с большим числом вычислительных ядер, реализующих *MPI* стандарт для обмена данными между параллельными вычислительными процессами и ориентирована на решение масштабных и сложных по геометрической композиции задач, требующих использования максимальных вычислительных ресурсов. В *LUCKY-A* для нейтронно-физических задач реализован алгоритм решения уравнения переноса Больцмана в рамках модели группового по энергии $P_m S_n$ приближения, где m – порядок разложения индикатрисы рассеивания по полиномам Лежандра, а n – порядок используемой угловой квадратуры.

Используемый алгоритм применяет метод параллельного решения уравнения переноса с использованием разбиения расчетной области на ряд подобластей, решение в которых вычисляется на параллельно работающих вычислительных ядрах суперкомпьютера [1–5]. Все модули программы *LUCKY-A* используют один и тот же геометрический модуль и одинаковую идеологию задания исходных данных. Геометрия расчета контролируется средствами *AUTOCAD* [6].

1. Параллельный алгоритм решения условно-критической задачи, основанный на методе пространственной декомпозиции

В программе *LUCKY-A* возможно использовать два параллельных алгоритма нахождения решения критической задачи. Первый – это классический алгоритм итерации источника деления [7], использующий метод распараллеливания по пространственным подобластям для нахождения решения уравнения с источником, второй – алгоритм итерации источника деления с распараллеливанием при получении решения, как по пространственным подобластям, так и по энергетическим группам [5]. Для первого алгоритма реализовано ускорение Гаусса–Зейделя для получения угловых потоков на внутренних итерациях. В данном ускорении все расчетные подобласти объединены в блоки. Каждый блок представляет собой группу из 8 соседних подобластей, имеющих общие граничные поверхности. Для каждой подобласти блока получается решение для углового потока в различных («встречных») октантах с последующим обменом решений на граничных поверхностях. Таким образом, входящий поток в каждую подобласть блока используется уже в текущей, а не в следующей итерации, что позволяет достигать выигрыша по числу итераций до 2–3 раз при проведении итерационного процесса. Второй алгоритм не имеет ускорения на внутренних итерациях.

Нахождение решения задачи по первому алгоритму:

решается уравнение переноса

$$L\varphi = \frac{1}{K} F\varphi + S\varphi, \tag{1}$$

где

$$L\varphi(x,y,z,\Omega,E) = \Omega_x \frac{\partial\varphi(x,y,z,\Omega,E)}{\partial x} + \Omega_y \frac{\partial\varphi(x,y,z,\Omega,E)}{\partial y} + \Omega_z \frac{\partial\varphi(x,y,z,\Omega,E)}{\partial z} + \Sigma_t\varphi(x,y,z,\Omega,E),$$

$$F\varphi(x,y,z,\Omega,E)=\frac{\chi(E)}{4\pi}\int\limits_{E}\int\limits_{\Omega}\nu\Sigma_f(x,y,x,E)\varphi(x,y,z,\Omega,E)\,d\Omega dE,$$

$$S\varphi(x,y,z,\Omega,E)=\int\limits_{E}\int\limits_{\Omega}P(x,y,z,\Omega'\to\Omega,\ E'\to E)\varphi(x,y,z,\Omega,E)d\Omega' dE'.$$

Итерационный процесс

$$L\varphi_{i+1}=\frac{1}{K_i}F\varphi_i+S\varphi_{i+1}, \tag{2}$$

где i – номер итерации, $k_i=\dfrac{Q\varphi_i}{Q\varphi_{i-1}}$, $Q=\int\limits_{V}\int\limits_{E}\int\limits_{\Omega}\nu\Sigma_f(x,y,z,E)\varphi(x,y,z,\Omega,E)\,dE\,d\Omega\,dV$,

$$K_i=\prod_{l=1}^{i}k_l \tag{3}$$

сходится $\varphi_i\to\varphi$ к решению задачи (1), а $K_i\to K$ собственному значению задачи (1). Решение уравнения (2) находится с помощью параллельного алгоритма нахождения решения по пространственным подобластям [1–4].

Нахождение решения задачи по второму алгоритму:

уравнение

$$L\varphi_{i+1}^{j}=\frac{1}{K_i}F\varphi_i+S\varphi_{i+1}^{j}+C\varphi_i \tag{4}$$

решается параллельно для каждой энергетической группы j с использованием параллельного алгоритма нахождения решения по пространственным подобластям. Оператор C описывает интеграл рассеивания, связывающий группу j с другими группами через процессы рассеивания в эту группу.

$$C\varphi_i=\sum_{l=1,l\neq j}^{N}\int\limits_{\Omega}P_{l\to j}(x,y,z,\Omega'\to\Omega)\varphi_i^{l}(x,y,z,\Omega')d\Omega',$$

где N – число энергетических групп.

Оператор S в уравнении (4) представляет только процесс внутригруппового рассеивания.

$$S\varphi_{i+1}^{j}=\int\limits_{\Omega}P_{j\to j}(x,y,z,\Omega'\to\Omega)\,\varphi_{i+1}^{j}(x,y,x,\Omega')d\Omega'.$$

После нахождения решения уравнения (4) оценивается K_{i+1} по формуле (3), затем получается новый источник $\dfrac{1}{K_{i+1}}F\varphi_{i+1}+C\varphi_{i+1}$ для уравнения (4) и продолжения итерационного процесса. Второй алгоритм требует большего числа обменов между параллельными процессами, тем не менее, связь между группами осуществляется (через интеграл деления и интеграл рассеивания в группу) интенсивнее, чем в первом алгоритме. В случае наличия полной матрицы рассеивания, предполагающей связь между всеми группами, этот алгоритм может быть более перспективным, чем первый.

Верификация алгоритмов решения условно-критической задачи, реализованных в программе, проводилась на известном тесте *C5G7* [8]. Тест предполагает расчет модельной композиции из четырех кассет с UO_2 и *MOX* топливом различного обогащения с поглощающими стержнями над ними. Расчет проводился с использованием семи энергетических групп с заданными макроконстантами. Помимо сравнения полученных значений $K_{\text{эфф}}$ для различных положений поглощающих стержней проводилось сравнение пространственных полей энерговыделения в кассетах.

К особенностям данного бенчмарка можно отнести необходимость детального описания геометрии расчетной области с целью максимального учета гетерогенности в системе, бенчмарк позиционируется авторами как гетерогенный. Размер пространственной сетки составлял ~ 0,12 см (полное число пространственных элементов в системе ~140 000 000), что позволяло детально описать

геометрию топливных ячеек. Расчеты проводились в S_4 приближении по угловой квадратуре с использованием 1000 вычислительных ядер, время решения задачи ~ 270 мин. Полученные результаты по *LUCKY-A* адекватно согласуются с результатами других авторов.

2. Параллельный алгоритм решения нестационарного уравнения переноса на базе параллельного решателя

Для решения нестационарного уравнения переноса Больцмана в программе *LUCKY-A* используется модуль *LUCKY_TD* [9]. В нем применяется алгоритм получения решения данной задачи на базе специально разработанного параллельного решателя для линейных систем уравнений первого порядка большой размерности, с использованием *MPI* технологии для обмена данными. Детальное описание алгоритма, реализованного в решателе, представлено в работе [10].

Для задачи на собственное значение $K_{\text{эфф}}$ находится как предел последовательности, полученный через итерационный процесс при решении уравнения

$$\frac{1}{\upsilon}\frac{\partial \varphi_i}{\partial t} = -(\Omega, \nabla \varphi_i) - \Sigma_t \varphi_i + \frac{\chi \iint \nu \Sigma_f \varphi_i d\Omega dE}{4\pi K_{\text{эфф},i-1}} + S\varphi_i. \qquad (5)$$

Если $Q = \int\limits_V \int\limits_E \int\limits_\Omega \nu \Sigma_f \varphi dE d\Omega dV$, тогда $k_i = \dfrac{Q\varphi_i}{Q\varphi_{i-1}}$, а $K_{\text{эфф}_i} = \prod\limits_{l=1}^{i} k_l$, i – номер временного шага, при начальных условиях φ_{i-1}. В результате итерационного процесса $\dfrac{\partial \varphi_i}{\partial t} \to 0$, а $\varphi_i \to \varphi$ к решению задачи (1), а $K_{\text{эфф},i} \to K_{\text{эфф}}$ собственному значению задачи (1). Получение решения (5) на каждом временном шаге i может быть рассмотрено как аналог внешней итерации при получении решения уравнения (1) по классическому алгоритму итерации источника. Отметим, что время для получения решения уравнения (1) по этому алгоритму больше времени получения решения по классическому стационарному алгоритму.

3. Исследование зависимости эффективности параллельного алгоритма от числа используемых вычислительных ядер суперкомпьютера для модельной условно-критической задачи

В качестве тестовой задачи была выбрана геометрическая композиция, представляющая из себя два куба имеющих один центр. Композиция имеет две материальные зоны. Внутренняя зона – куб с ребром 120 см – это делящийся материал (типичный для быстрых реакторов БН), а внешняя зона – куб с ребром 140 см – это материал отражателя. Использовались 21 групповые ядерные макроконстанты в P_1 приближении в разложении по индикатрисе рассеивания.

Расчеты проводились в P_1S_4 приближении по программе *LUCKY-A*, кроме того, получено решение данной задачи по программе *TORT* [11], реализующей метод дискретных ординат, и программе *MMKKENO* [12] методом Монте-Карло. Рассчитывалась 1/8 часть системы (с учетом симметрии решаемой задачи) с граничными условиями отражения слева, по фронту и снизу системы. К остальным границам системы применялось свободное граничное условие (ноль влета частиц в систему). Таким образом, геометрические размеры системы составляли $70 \cdot 70 \cdot 70$ см по координатным осям *X*, *Y*, *Z*. Размерность пространственной сетки соответственно по осям $70 \cdot 70 \cdot 70$, шаг сетки равнялся 1 см. Таким образом, в отражателе использовались 10 пространственных интервалов. Так как использовались константы для реактора БН, в данной задаче отсутствуют переходы

вверх из группы в группу. Это особенность данной задачи. Относительная точность полученного решения для внутренних итераций $\varepsilon = 10^{-8}$, а относительная точность $K_{\text{эфф}}$, $\varepsilon_k = 10^{-6}$.

Введем величину эффективности параллельного процесса, определив ее как

$$E_p = t_1/(t_n n), \tag{6}$$

где t_1 – время решения задачи на одном вычислительном ядре, t_n – время решения задачи с использованием n вычислительных ядер. E_p определяет выигрыш по времени на одно используемое вычислительное ядро.

Известная формула Амдала $E = \dfrac{T_1}{\alpha T_1 + (1-\alpha)T_1/n} = \dfrac{1}{\alpha + (1-\alpha)/n}$, определяющая выигрыш по времени, при использовании n параллельных вычислений, показывает, что если в алгоритме есть последовательная часть, доля времени которой α, то в пределе при $n \to \infty$ полученное ускорение не может превысить величину $1/\alpha$. Решение практических задач обычно занимает значительное время. Последовательная часть программы чаще всего определяется временем ввода исходных данных, которое составляет тысячные или сотые доли от времени расчета, что дает возможность использовать оценку E_p для эффективности и получать значительный выигрыш по времени при использовании алгоритмов с хорошей эффективностью параллельного процесса.

3.1. Результаты

Результаты расчетов представлены в таблицах. В таблицах дана информация по разбиению расчетной области на подобласти по пространственным осям (с равномерным шагом) для каждого варианта расчета, эффективность параллельного процесса нахождения решения, доля времени на расчет (без учета времени обмена данных), полное число необходимых внутренних и внешних итераций для каждого варианта, время расчета.

Описание параметров таблицы:

$N = Nx \cdot Ny \cdot Nz$ – число используемых ядер,

Nx – число равномерных разбиений расчетной области на подобласти по оси *X*,

Ny – число равномерных разбиений расчетной области на подобласти по оси *Y*,

Nz – число равномерных разбиений расчетной области на подобласти по оси *Z*,

Nx, Ny, Nz – подбирались так, чтобы полное число интервалов по осям *X*, *Y*, *Z* делилось нацело,

E_p – эффективность параллельного процесса,

T – время расчета в минутах,

I – число внутренних итераций в расчете,

O – число внешних итераций в расчете,

P – доля времени затраченного на вычисления от полного времени расчета.

Таблица 1

Результаты расчетов с пространственным распараллеливанием. $K_{эфф}$ = 0,98845

N	1	50	125	175	245	343	490	700	1000
Nx	1	5	5	7	7	7	10	10	10
Ny	1	5	5	5	7	7	7	10	10
Nz	1	2	5	5	5	7	7	7	10
E_p	1	0,8	0,82	0,79	0,76	0,67	0,55	0,40	0,30
T	485	12	4,7	3,5	2,6	2,1	1,8	1,71	1,58
I	16666	20905	21737	22276	22774	23530	25669	26789	27200
O	22	22	22	22	22	22	22	22	22
P	0,99	0,94	0,94	0,93	0,90	0,83	0,75	0,58	0,45

Таблица 2

Результаты расчетов с пространственно-энергетическим распараллеливанием. $K_{эфф}$ =0,98844. $N = Nx \cdot Ny \cdot Nz \cdot Ng$, где Ng – число энергетических групп

N	21	42	84	168	420	1050
Nx	1	2	2	2	5	5
Ny	1	1	2	2	2	5
Nz	1	1	1	2	2	2
E_p	0,208	0,188	0,174	0,166	0,141	0,08
T	111	61,2	33,1	17,3	8,15	5,82
I	3978	4463	4848	5057	5424	5913
O	87	87	87	87	87	87
P	0,94	0,94	0,96	0,93	0,84	0,51

Результат расчета по *MMKKENO* $K_{эфф} = 0{,}9895 \pm 0{,}0001$. Результат расчета по *TORT* $K_{эфф} = 0{,}98906$, относительная погрешность *TORT* и *LUCKY-A* $\frac{\delta k}{k} \approx 6 \cdot 10^{-4}$. Отметим, что при расчетах по разным программам, использовались одни и те же макроконстанты.

Полная пространственная сетка во всех расчетах сохранялась. В зависимости от числа разбиений на пространственные подобласти менялась пространственная сетка на вычислительном ядре, с увеличением числа вычислительных ядер (числа подобластей) она уменьшалась. Из таблиц видно, что при увеличении числа разбиений на подобласти наблюдается увеличение числа необходимых внутренних итераций (до 1,6 раз при использовании 1000 ядер) для получения решения, так как для связи между подобластями с увеличением их числа, необходимо большее число итераций для того, чтобы удаленные подобласти могли получить связь между собой. Это приводит к снижению эффективности параллельного процесса. При уменьшении вычислительной нагрузки на ядро (для числа ядер больше 500) наблюдается снижение доли времени на вычисления, что связано с тем, что время на обмен данными становится сравнимо с временем расчета итерации. Это тоже способствует снижению эффективности параллельного процесса с увеличением числа пространственных подобластей.

Немонотонность эффективности от числа используемых вычислительных процессов в табл. 1 обусловлена особенностями архитектуры суперкомпьютера, а именно последовательностью загрузки процессов на вычислительные модули, что может приводить к колебаниям времени расчета за счет различного времени обмена данными между процессами.

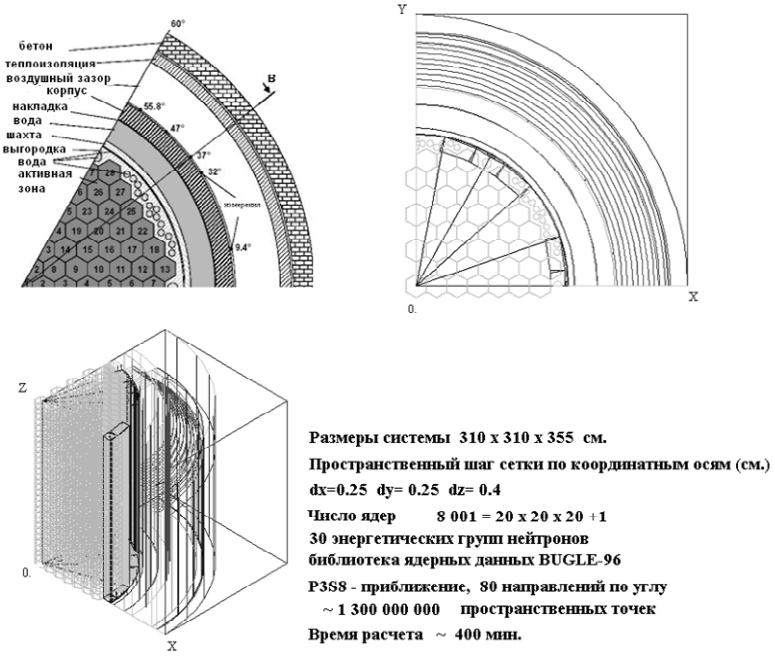
Наблюдается значительное увеличение числа внешних итераций во втором алгоритме, и, как следствие, снижение эффективности параллельного процесса для всех расчетных композиций по сравнению с первым алгоритмом. Следует отметить, что второй алгоритм с использованием пространственно-энергетического распараллеливания оказался значительно менее эффективным, чем первый алгоритм. Меньшая эффективность второго алгоритма может быть обусловлена как отсутствием ускорения для него, так и особенностью решаемой задачи, а именно, отсутствием рассеивания вверх из группы в группу. Данный алгоритм может быть наиболее эффективен для задач с полной матрицей рассеивания для групповых сечений.

4. Практическое применение

Основное применение программы *LUCKY-A* – решение задач защиты от нейтронного и гамма излучения в больших и сложных по геометрической композиции системах. Как уже отмечалось, программа разработана для использования максимальных ресурсов суперкомпьютера. Приведем пример такого использования при решении задачи нахождения потока быстрых нейтронов на корпус атомной станции ВВЭР Балаково. На рисунке представлена геометрия задачи. При решении задачи используется большая пространственная сетка $\sim 10^9$ пространственных элементов, с раз-

мерностью сетки в несколько миллиметров, что позволяет очень детально описать геометрию расчета, P_3S_8 приближение и 30 энергетических групп. Источник в активной зоне задавался с учетом потвэльного распределения посредством интерфейса с программой расчета активной зоны реактора ПЕРМАК. Объем расчетной области ~35 м^3. Число используемых вычислительных ядер для решения ~ 8 000. Время решения задачи составляет несколько часов.

Атомная станция ВВЭР Балаково, расчет потока быстрых нейтронов на корпус реактора, расчет проводился 8 001 ядрах на вычислительном комплексе НИЦ "Курчатовский Институт"



Геометрия и параметры расчетной модели для задачи расчета потока быстрых нейтронов для атомной станции ВВЭР Балаково

Заключение

Проведенные расчеты тестовой композиции с различным числом разбиений по пространственным подобластям показали достаточную эффективность использования первого параллельного алгоритма решения критической задачи (1), реализованного в программе *LUCKY-A*, даже на достаточно большом числе вычислительных ядер. Чем более нагружено вычислительное ядро (время расчета итерации больше времени обмена данными), тем меньше доля времени затраченного на обмен данными, тем выше эффективность параллельного процесса. Например, в варианте для 1000 ядер в табл. 1 в разбивке по координатным осям $10 \cdot 10 \cdot 10$ доля расчетного времени $P = 0{,}45$ при эффективности $E_p = 0{,}3$. В случае идеального суперкомпьютера (нет потерь времени на обмен данными) эффективность могла быть $E = E_p / P = 0{,}3/0{,}45 = 0{,}66$.

Использование огромных ресурсов суперкомпьютера при проведении расчетов позволяет применять, если это действительно необходимо, детальную пространственную сетку, которую затруднительно использовать в расчетах на персональном компьютере. Например, решение рассмотренной задачи с сеткой $140 \cdot 140 \cdot 140$ на 1000 ядрах занимает менее 7 минут, несмотря на то, что число пространственных интервалов порядка 3 000 000. Суперкомпьютер также необходим при проведении быстрых расчетов, когда есть жесткие временные критерии для получения результата.

Все расчеты проводились на суперкомпьютере МВС-10П в Межведомственном Суперкомпьютерном Центре Российской Академии Наук [13]. Автор выражает благодарность МСЦ за возможность проведения расчетов и выделенные ресурсы.

Литература

1. Моряков А. В. Программа *LUCKY*. Решение уравнения переноса нейтронов и гамма излучения с использованием параллельных технологий // Вопросы атомной науки и техники. Сер. Физика ядерных реакторов. 2010. Вып. 4. С. 18–29.

2. Моряков А. В. Результаты расчетов по программе *LUCKY*. Сравнение с другими программами и экспериментальными данными // Вопросы атомной науки и техники. Сер. Физика ядерных реакторов. 2010. Вып. 4. С. 30–40.

3. Моряков А. В. Алгоритм получения угловых потоков в ячейке для многопроцессорных программ *LUCKY* и *LUCKY_C* // Вопросы атомной науки и техники. Сер. Физика ядерных реакторов. 2011. Вып. 1. С. 3–7.

4. Moryakov A. V. The *LUCKY* program for solving the transport equation for neutrons and gamma radiation with the use of parallel technologies // Physics of Atomic Nuclei. 2011. Vol. 74, N 14. P. 1891.

5. Моряков А. В. Алгоритм пространственно-энергетического распараллеливания для нахождения решения критической задачи, реализованный в многопроцессорной программе *LUCKY_C* // Вопросы атомной науки и техники. Сер. Физика ядерных реакторов. 2012. Вып. 1. С. 24–27.

6. Красковский Д. Г., Виноградов А. В. AutoCAD 2000 для всех. М.: Компьютер Пресс, 1999.

7. Шишков Л. К. Методы решения диффузионных уравнений двухмерного ядерного реактора. М.: Атомиздат, 1976.

8. Ковалишин А. А., Моряков А. В. Расчет по программе *LUCKY-A* бенчмарка *C5G7* (в печати) // Вопросы атомной науки и техники. Сер. Физика ядерных реакторов. 2016.

9. Моряков А. В. Программа *LUCKY_TD* для решения нестационарного уравнения переноса с использованием параллельных вычислений // Вопросы атомной науки и техники. Сер. Физика ядерных реакторов. 2015. Вып. 2. С. 15–19.

10. Моряков А. В. Алгоритм решения линейной задачи Коши для систем обыкновенных дифференциальных уравнений большой размерности с использованием параллельных вычислений // Вопросы атомной науки и техники. Сер. Физика ядерных реакторов. 2015. Вып. 2. С. 4–14.

11. Doors 3.2: One,Two and Three-Dimensional Discrete Ordinates Neutron/Photon Transport Code System. RSIC Code Package, CCC-650.

12. Блыскавка А. А., Мантуров Г. Н., Николаев М. Н., Цибуля А. М. Аннотация программного комплекса *MMKKENO*: Препринт № 3145. Обнинск: ФЭИ, 2009.

13. [Electronic resource]. Mode of Access: https://www.jscc.ru.