

Методы распараллеливания на многопроцессорных системах с распределенной памятью алгоритмов решения уравнения переноса излучения в программе РАДУГА*

О.В.Николаева, Л.П.Басс, В.С.Кузнецов

Рассмотрены алгоритмы распараллеливания вычислений, используемые в программе Радуга-5.2(П) при решении уравнения переноса нейтральных частиц в 1D, 2D, 3D областях методом дискретных ординат на многопроцессорных системах с распределенной памятью. Приведены результаты тестирования эффективности алгоритмов по времени и памяти на примере реальной задачи радиационной защиты.

1. Введение

Уравнение переноса излучения моделирует взаимодействие потоков частиц различного типа (нейтронов, фотонов, электронов) с веществом и широко используется в многочисленных приложениях. В частности, с его помощью моделируются поля нейтронов и гамма-квантов в защитах ядерных реакторов и в фантоме человека (задачи медицинской физики и биомедицины), а также поля рассеянных фотонов в атмосфере и облаках (задачи атмосферной оптики).

Основные трудности, возникающие при численном решении уравнения переноса, связаны со следующими обстоятельствами.

1. Искомая функция - плотность потока частиц – определяет в общем случае распределение частиц по пространственным координатам, скоростям и времени, то есть зависит от 7 переменных (от 6-ти переменных в стационарном случае).
2. Уравнение переноса, учитывающее процессы не только поглощения частиц средой, но и рассеяния и деления, является интегро-дифференциальным, не разрешимым явно.
3. Область расчета может иметь большие размеры в длинах свободного пробега частицы.
4. Коэффициенты, описывающие процесс взаимодействия частиц с веществом, представляют собой сильно нерегулярные функции, связывающие энергию вылетающей частицы с энергией налетающей частицы и углом ее рассеяния.
5. Источник излучения в стационарных задачах часто является сингулярным - занимающим очень малую область пространства или/и испускающим частицы в узком интервале направлений (стационарный лазер, поток солнечного излучения, пучок выводимых из ускорителя или активной зоны реактора частиц).
6. В нестационарных задачах источник может быть еще и импульсным (лазер, излучающий в узком временном интервале).

Эти обстоятельства вместе с высокими требованиями к точности решения задач о переносе излучения приводят к необходимости разработки сложных трудоемких алгоритмов, учитывающих особенности коэффициентов и источников. Приемлемого времени счета, особенно при решении нестационарных, обратных задач и задач оптимизации, в этом случае можно достичь лишь на суперкомпьютерах с параллельной архитектурой.

Программа Радуга-5.2(П) [1] была создана для решения уравнения переноса нейтральных частиц (нейтронов, фотонов) в 1D, 2D, 3D областях методом дискретных ординат [2], [3] на многопроцессорных компьютерах (раздел 2). Программа написана на языке Fortran-77, распараллеливание вычислений выполнено с помощью команд языка MPI. Используемые в программе алгоритмы распараллеливания (раздел 3) уменьшают как время счета, так и объем требуемой памяти. Тестирование эффективности параллельных алгоритмов по времени и памяти выполнено на реальной задаче расчета защиты реактора ВВЭР-1000 в двумерной

* Работа выполнена при финансовой поддержке Программы №3 фундаментальных исследований Отделения Математических наук РАН.

цилиндрической геометрии (раздел 4.1) и задаче расчета матриц рассеяния солнечного света в морской воде (раздел 4.2).

2. Метод дискретных ординат

Стационарные многогрупповые уравнения переноса излучения опираются на разбиение интервала кинетической энергии частицы $(0, E_{\max})$ на Q групп (энергетических интервалов)

$$0 < E_{Q+1/2} < \dots < E_{q+1/2} < E_{q-1/2} < \dots < E_{1/2} = E_{\max},$$

каждой из которых соответствует средняя энергия $E_q \in (E_{q+1/2}, E_{q-1/2})$. Уравнения имеют следующий вид

$$\mathcal{L}\Psi = \mathcal{S}\Psi + \mathcal{D}\Psi + F, \quad \vec{r} \in G, \quad \vec{\omega} \in \Omega, \quad q = 1, \dots, Q, \quad (1)$$

где линейные операторы \mathcal{L} , \mathcal{S} , \mathcal{D} определяются соотношениями

$$\begin{aligned} \mathcal{L}\Psi &= \vec{\omega} \vec{\nabla} \Psi_q(\vec{r}, \vec{\omega}) + \sigma_{t,q}(\vec{r}) \Psi_q(\vec{r}, \vec{\omega}), \\ \mathcal{S}\Psi &= \sum_{p=1}^q \int_{\Omega} \sigma_{s,p \rightarrow q}(\vec{r}, \vec{\omega}, \vec{\omega}') \Psi_p(\vec{r}, \vec{\omega}') d\vec{\omega}', \\ \mathcal{D}\Psi &= \sum_{p=1}^Q \sigma_{f,p \rightarrow q}(\vec{r}) \int_{\Omega} \Psi_p(\vec{r}, \vec{\omega}') d\vec{\omega}'. \end{aligned} \quad (2)$$

Здесь: решение $\Psi_q(\vec{r}, \vec{\omega})$ - плотность потока частиц - число частиц с кинетической энергией из группы q , имеющих направление из единичного телесного угла в окрестности выбранного направления $\vec{\omega}$, за единицу времени проходящих через единичную площадку, расположенную в точке \vec{r} перпендикулярно вектору $\vec{\omega}$.

Каждому единичному вектору $\vec{\omega}$ отвечает точка на единичной сфере Ω , задаваемая в сферической системе координат углами (θ, φ) , см. **Рис. 1**. Оператор \mathcal{S} в правой части уравнения (1) – сумма интегралов по сфере Ω - описывает замедление частиц (переход из группы с меньшим номером (и с большей энергией) в группу с большим номером (и с меньшей энергией) при их взаимодействии со средой; оператор \mathcal{D} описывает рождение новых частиц при делении ядер. Функция $F_q(\vec{r}, \vec{\omega})$ есть не зависящий от решения источник частиц.

Уравнение переноса (1) содержит следующие коэффициенты

- o $\sigma_{t,q}(\vec{r})$ – полное сечение взаимодействия с веществом частиц из q -й группы,
- o $\sigma_{f,p \rightarrow q}(\vec{r})$ - сечение деления, при котором захватываемая частица принадлежит группе p , а образовавшаяся при распаде ядра частица - группе q ,
- o $\sigma_{s,p \rightarrow q}(\vec{r}, \vec{\omega}, \vec{\omega}')$ – сечение рассеяния, при котором частица переходит из группы p в группу q , и направление ее движения меняется с $\vec{\omega}'$ на $\vec{\omega}$ (это сечение в каждой пространственной точке \vec{r} зависит от косинуса угла рассеяния – скалярного произведения единичных векторов $\vec{\omega}, \vec{\omega}'$).

Пространственная область G считается выпуклой, а на ее границе ∂G задано значение входящего в область G потока частиц

$$\Psi_q(\vec{r}, \vec{\omega}) \Big|_{\vec{\omega} \cdot \vec{n}(\vec{r}) < 0} = \mathcal{R}\Psi_q = \int_{\vec{\omega}' \cdot \vec{n}(\vec{r}) > 0} \Psi_q(\vec{r}, \vec{\omega}') R(\vec{r}, \vec{\omega}, \vec{\omega}') d\vec{\omega}' \vec{\omega} \cdot \vec{n}(\vec{r}), \quad (3)$$

где $\vec{r} \in \partial G$, $\vec{n}(\vec{r})$ – внешняя нормаль к поверхности ∂G в точке \vec{r} , $\vec{\omega} \cdot \vec{n}(\vec{r})$ – косинус угла между внешней нормалью $\vec{n}(\vec{r})$ и направлением переноса $\vec{\omega}$, $R(\vec{r}, \vec{\omega}, \vec{\omega}')$ – функция отражения. Условие (3) включает в себя различные условия отражения частиц от внешней границы области (в частности, при $R(\vec{r}, \vec{\omega}, \vec{\omega}') = 0$ получаем нулевое краевое условие).

Решение задачи (1)-(3) сеточным методом дискретных ординат опирается на введение сеток по пространственным переменным и направлениям переноса. Сетка по направлениям вводится разбиением единичной сферы (каждой точке которой сопоставлен единичный вектор переноса $\vec{\omega}$) на M подобластей, каждой из которых отвечает узел $\vec{\omega}_m$ и вес (площадь подобласти) $\Delta\omega_m$, см. **Рис. 1**.

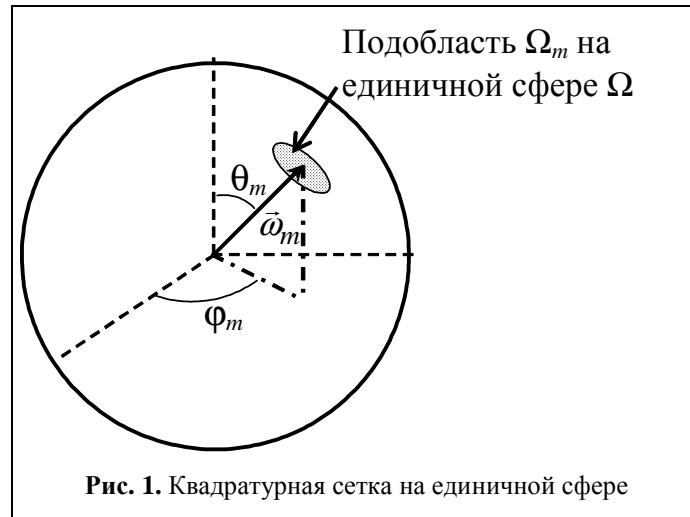
Важным является выбор способа аппроксимации угловой зависимости сечений рассеяния $\sigma_{s,p \rightarrow q}(\vec{r}, \vec{\omega} \cdot \vec{\omega}')$. В первом способе сечения аппроксимируются разложением по полиномам Лежандра $P_\ell(\xi)$

$$\sigma_{s,p \rightarrow q}(\vec{r}, \vec{\omega} \cdot \vec{\omega}') = \sum_{\ell=0}^L (2\ell+1) P_\ell(\vec{\omega} \cdot \vec{\omega}') \sigma_{s,p \rightarrow q, \ell}(\vec{r}), \quad (4)$$

где $\sigma_{s,p \rightarrow q, \ell}(\vec{r}) = \frac{1}{2} \int_{-1}^1 d\xi \sigma_{s,p \rightarrow q, \ell}(\vec{r}, \xi) P_\ell(\xi)$ – моменты разложения. Это приводит к разложению операторов рассеяния \mathfrak{S} по сферическим функциям $Y_{\ell, m}(\vec{\omega})$:

$$\mathfrak{S}\Psi = \sum_{p=1}^q \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} Y_{\ell, m}(\vec{\omega}) \sigma_{s,p \rightarrow q, \ell}(\vec{r}) M_{\ell, m}[\Psi_p],$$

$$M_{\ell, m}[\Psi_p] = \int_{\Omega} Y_{\ell, m}(\vec{r}, \vec{\omega}') \Psi_p(\vec{r}, \vec{\omega}') d\vec{\omega}',$$



и расчет интеграла в (2) сводится к расчету моментов $M_{\ell, m}[\Psi_p]$ по квадратурным формулам

$$M_{\ell, m}[\Psi_p] = \sum_{m=1}^M Y_{\ell, m}(\vec{r}, \vec{\omega}_m) \Psi_p(\vec{r}, \vec{\omega}_m) \Delta\omega_m.$$

Здесь $\Psi_p(\vec{r}, \vec{\omega}_m)$ – значения потока в узлах квадратуры.

Во втором способе зависимость сечения рассеяния $\sigma_{s,p \rightarrow q}(\vec{r}, \vec{\omega} \cdot \vec{\omega}')$ от косинуса угла рассеяния $\vec{\omega} \cdot \vec{\omega}'$ задается значениями на некоторой сетке. Тогда оператор \mathfrak{S} заменяется суммой

$$\mathfrak{S}\Psi_q(\vec{r}, \vec{\omega}_n) = \sum_{p=1}^q \sum_{m=1}^M \int_{\Omega} s_{p \rightarrow q, n, m}(\vec{r}) \Psi_p(\vec{r}, \vec{\omega}_m) \Delta\omega_m. \quad (5)$$

Величины $s_{p \rightarrow q, n, m}(\vec{r})$ определяются как интегралы от сечений рассеяния $\sigma_{s,p \rightarrow q}(\vec{r}, \vec{\omega}_n \cdot \vec{\omega}')$ по отвечающей узлу $\vec{\omega}_m$ подобласти единичной сферы, см. Рис. 1

$$s_{p \rightarrow q, n, m}(\vec{r}) = \frac{1}{\Delta\omega_m} \int_{\Omega_m} \sigma_{s,p \rightarrow q}(\vec{r}, \vec{\omega}_n \cdot \vec{\omega}') d\omega'. \quad (6)$$

Они образуют матрицу рассеяния; каждый ее элемент $s_{p \rightarrow q, n, m}(\vec{r})$ отвечает переходу частицы при акте рассеяния из группы p в группу q и из направления $\vec{\omega}_m$ в направление $\vec{\omega}_n$.

3. Методы распараллеливания

3.1 Метод пространственных подобластей

Метод пространственных подобластей в решении задачи (1)-(3) опирается на разбиение области расчета G на подобласти. В программе Радуга-5.2(П) разбиение выполняется поверхностями, параллельными координатным плоскостям, см. Рис. 2. Каждая подобласть рассчитывается своим процессором по следующей итерационной схеме

$$\mathfrak{L}\Psi^{i+1, j+1} = \mathfrak{S}\Psi^{i, j+1} + \mathfrak{D}\Psi^{i, j} + F, \quad \vec{r} \in G, \quad \vec{\omega} \in \Omega, \quad q = 1, \dots, Q,$$

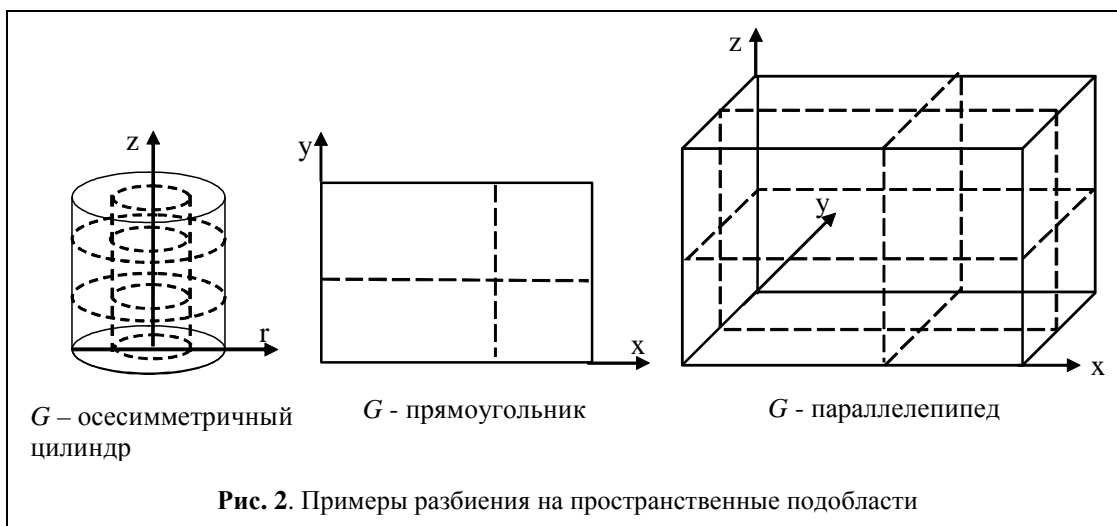
$$\Psi_q^{i+1, j+1}(\vec{r}, \vec{\omega}) \Big|_{\vec{\omega} \cdot \vec{n}(\vec{r}) < 0} = \mathfrak{R}\Psi_q^{i, j+1}.$$

Здесь используется два итерационных цикла. Внешнему циклу отвечает индекс j , и для каждого шага внешнего цикла выполняется внутренний цикл по индексу i . Для каждой внешней итерации известными с предыдущего итерационного шага считаются значения оператора \mathfrak{D} , описывающего процесс деления ядер. На каждой внутренней итерации известными считаются значения оператора \mathfrak{S} (описывающих процессы рассеяния и замедления частиц при взаимодействии с нуклидами), \mathfrak{R} (отвечающего за отражение частиц от границы подобласти) и \mathfrak{D} (с предыдущей внешней итерации). Если вещества области расчета не содержат делящихся ядер, внешний итерационный цикл не совершается.

Таким образом, реально на каждом итерационном шаге в каждой подобласти решается следующая задача

$$\mathfrak{L}\Psi_q(\vec{r}, \vec{\omega}) = Q_q(\vec{r}, \vec{\omega}), \quad \vec{r} \in G, \quad \vec{\omega} \in \Omega, \quad q = 1, \dots, Q, \quad \Psi_q(\vec{r}, \vec{\omega}) \Big|_{\vec{\omega} \cdot \vec{n}(\vec{r}) < 0} = \phi_q(\vec{r}, \vec{\omega}),$$

где внутренний источник $Q_q(\vec{r}, \vec{\omega})$ известен из расчета той же подобласти на предыдущей итерации, а граничный источник $\phi_q(\vec{r}, \vec{\omega})$ получен расчетом на предыдущей итерации соседней подобласти. Поэтому после каждого итерационного шага производится обмен значениями потоков Ψ на границах между процессорами, отвечающими соседним подобластям. Итерационный процесс считается сошедшимся, если изменение потоков на соседних итерациях (внешних и внутренних) становится меньше априори заданного числа.



Несомненным преимуществом данного алгоритма является его эффективность по памяти – каждый процессор хранит в своей оперативной памяти только те величины, которые используются при расчете прикрепленной к нему подобласти. Эффективность алгоритма по времени тем выше, чем меньше времени занимают межпроцессорные обмены и чем больше уменьшается время счета при увеличении числа процессоров. Результаты тестирования эффективности этого алгоритма по времени и по памяти на примере расчета модели защиты реактора ВВЭР-1000 представлены в разделе 4.1.

3.2 Метод расчета элементов матриц рассеяния

В программу Радуга-5.2(П) также введено дополнительное распараллеливание тех трудоемких шагов алгоритма расчета, которые являются идентичными для всех подобластей. Таким трудоемким шагом является расчет матриц рассеяния, аппроксимирующих интегральный оператор \mathcal{E} в правой части уравнения (1), см. (5). Такой расчет выполняется по формулам (6) с помощью равномерной дополнительной квадратуры, вводимой на подобласти интегрирования, см. **Рис. 1**, и требует больших временных затрат.

В программе Радуга-5.2(П) расчет матриц рассеяния, используемых на текущем внутреннем итерационном цикле, выполняется перед началом этого цикла и эти матрицы хранятся в оперативной памяти. При этом в оперативной памяти каждого процессора хранятся только матрицы тех материалов, которые присутствуют в отвечающей процессору пространственной подобласти. Расчет матриц выполняется параллельным образом – элементы каждой матрицы делятся на N групп (N – число процессоров), каждый процессор вычисляет свою группу элементов каждой матрицы; полная матрица формируется после межпроцессорных обменов. Результаты тестирования эффективности этого алгоритма по времени на примере расчета матрицы рассеяния света в морской воде представлены в разделе 4.2.

4. Численное тестирование эффективности алгоритмов

В этом разделе представлены результаты тестирования эффективности алгоритмов программы Радуга-5.2(П). Используются две функции-критерия

$$\begin{aligned} E_{\text{time}}(N) &= T(1)/[T(N) N] - \text{эффективность по времени,} \\ E_{\text{memory}}(N) &= M(1)/[M(N) N] - \text{эффективность по памяти.} \end{aligned} \quad (7)$$

Здесь N – число процессоров, $T(N)$ – время расчета задачи на N процессорах, $M(N)$ – средний размер памяти, занимаемый одним процессором при счете на N процессорах. Очевидно, что в идеальном случае каждая из эффективностей (7) должна быть равна единице.

В реальных случаях эффективности оказываются меньше единицы из-за неизбежных потерь, связанных с распределением величин и вычислений между процессорами. Например, эффективность по времени снижается из-за потерь на межпроцессорные обмены и из-за того,

что часть вычислений дублируется на всех процессорах. Эффективность по памяти снижается, поскольку часть массивов является общей для всех подобластей.

Повысить эффективность расчетов можно, учитывая структуру используемой ЭВМ. Мы проводили расчеты на суперкомпьютере МВС-100К (Межведомственный суперкомпьютерный центр Российской Академии Наук, www.jssc.ru), решающее поле которого состоит из 990 вычислительных модулей. Каждый из них содержит

- два четырёхядерных микропроцессора Intel Xeon, работающих на частоте 3 ГГц;
- оперативную память DDR2 объёмом не менее 4 Гбайт;
- жёсткий диск объёмом не менее 36 Гбайт;
- интерфейсную плату HP Mezzanine Infiniband DDR;
- два интегрированных контроллера Gigabit Ethernet.

Вычислительные модули связаны между собой высокоскоростной коммуникационной сетью Infiniband DDR, транспортной и управляющей сетями Gigabit Ethernet.

Поскольку каждый модуль содержит 8 микропроцессоров, в настоящее время общее число микропроцессоров на МВС-100К равно 7920.

Поскольку микропроцессоры, относящиеся к одному вычислительному модулю, имеют общую оперативную и дисковую память, то при решении на таком суперкомпьютере задач, требующих использования больших объемов памяти на каждом процессоре, целесообразно использовать микропроцессоры, принадлежащие разным модулям, чтобы избежать конкуренции между ними за память и тем самым не допустить увеличения времени счета при увеличении числа процессоров. Однако при этом неизбежны временные затраты на межмодульные обмены, скорость которых ниже, чем скорость обмена данными между микропроцессорами одного модуля. Отметим, что реальные задачи для уравнения переноса всегда требуют больших объемов памяти, из-за необходимости хранить значения потоков во всех пространственных точках во всех группах, поскольку они используются при вычислении входящих в правую часть уравнения (1) операторов \mathcal{S} и \mathcal{D} .

При решении задач, не требующих большой оперативной памяти, целесообразнее использовать все микропроцессоры модуля, чтобы снизить время межпроцессорных обменов.

Далее представлены результаты исследования эффективности алгоритмов распараллеливания для задач двух этих типов.

4.1 Расчет защиты реактора ВВЭР-1000

Часто при анализе эффективности распараллеливания рассматривают небольшие модельные задачи. Мы рассматриваем реальную задачу - модель защиты реактора ВВЭР-1000. Реактор здесь представлен как сильно гомогенизированный осесимметричный цилиндр высотой 572.42 см и радиусом 365.51 см. Источник нейтронов - активная зона, прилегающая к оси цилиндра, занимает 11.6 % области расчета. Геометрия защиты представлена 28 различными материалами (вода, воздух, сталь, железо, бетон, железобетон различного типа). Рассматривалась задача замедления нейтронов – сечения деления считались равными нулю.

Расчет поля быстрых нейтронов (с энергией более 1Мэв) был выполнен на регулярной пространственной сетке, состоящей из 26815 ячеек с использованием квадратуры, содержащей 84 узла, равномерно распределенных на единичной полусфере.

Угловая зависимость сечений рассеяния аппроксимировалась восьмью полиномами Лежандра (P_7 приближение – в разложении (4) параметр $L=7$) или средними значениями сечений в 100 интервалах равномерной сетки по интервалу изменения косинуса угла рассеяния $[-1, 1]$ (A_{100} приближение).

В рассматриваемой задаче требования к объему памяти велики (требуется 74.65 Мб для P_7 приближения, 218.02 Мб для A_{100} приближения при расчете всей задачи на одном процессоре). При увеличении числа процессоров рабочие массивы равномерно распределяются между процессорами и требования к памяти одного процессора существенно снижаются.

При решении этой задачи на суперкомпьютере МВС-100К в одном модуле выбирался один процессор. Значения эффективностей (7) представлены в следующей таблице.

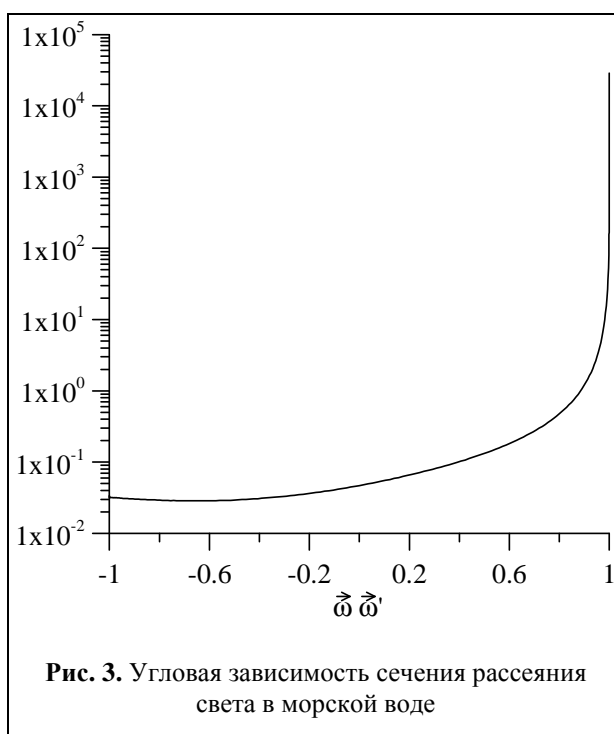
Таблица 1. Эффективности параллельных алгоритмов в задаче расчета защиты реактора ВВЭР-1000

Число процессоров	P ₇ приближение		A ₁₀₀ приближение	
	$E_{time}(N)$	$E_{memory}(N)$	$E_{time}(N)$	$E_{memory}(N)$
1	1	1	1	1
4	0.7	1.02	0.9	0.91
9	0.74	0.98	0.82	0.76
16	0.58	0.93	0.53	0.66
25	0.53	0.85	0.45	0.55
36	0.45	0.76	0.29	0.44
49	0.36	0.66	0.28	0.32

Снижение эффективности по памяти с увеличением числа процессоров обусловлено тем, что в программе большие объемы массивов (сеток, сечений, матриц рассеяния) являются общими для всех пространственных подобластей.

Снижение эффективности по времени происходит из-за увеличения времени межпроцессорных обменов, происходящем при увеличении числа процессоров. При этом, поскольку в A₁₀₀ варианте процессоры обмениваются не только граничными значениями потоков (как в P₇ варианте), но и элементами матриц рассеяния для всех материалов и всех переходов $p \rightarrow q$, эффективность по времени в A₁₀₀ приближении меньше, чем в P₇ приближении.

4.2 Расчет матриц рассеяния света в морской воде



Анализ эффективности расчета матриц рассеяния (без расчета потоков) был выполнен для сечения рассеяния света в морской воде (индикатриса Мобли [4]), см. Рис. 3. Такие сильно вытянутые сечения не могут быть аппроксимированы с хорошей точностью небольшим числом полиномов Лежандра и в расчетах по программе Радуга-5.2(П) представляются в дискретной форме, что приводит к аппроксимации интеграла рассеяния матрицей рассеяния, см. формулу (5).

Расчет матрицы рассеяния для квадратуры, содержащей $M=480$ узлов на полусфере, был выполнен на суперкомпьютере МВС-100К при использовании всех свободных процессоров каждого модуля. Общее число элементов матрицы равно $M^2 = 230400$, число узлов

дополнительной квадратуры, используемой при расчете каждого элемента, было выбрано равным 144. Полученная эффективность по времени представлена в Таблице 2.

Таблица 2. Эффективность параллельных алгоритмов в задаче расчета матрицы рассеяния

Число процессоров	$E_{\text{time}}(N)$
1	0.98
4	0.96
15	1.01
25	0.94
50	0.898
100	0.889

Высокая эффективность распараллеливания в этой задаче обусловлена небольшими требованиями к памяти в этой задаче и небольшим числом межпроцессорных обменов.

Заключение

Исследование эффективности алгоритмов распараллеливания программы Радуга-5.2(П) решения уравнения переноса излучения в многомерных областях методом дискретных ординат выполнено на реальной задаче радиационной защиты, требующей больших объемов оперативной памяти. Хотя в таких задачах эффективность алгоритмов распараллеливания близка к единице лишь на малом числе процессоров, использование большого числа процессоров позволяет снизить время расчета на порядок и более. Это позволяет применять при решении сложных реальных задач о переносе излучения трудоемкие алгоритмы, которые невозможно выполнить за разумное время на однопроцессорных компьютерах.

Литература

1. Басс Л.П., Гермогенова Т.А., Николаева О.В., Кузнецов В.С. Радуга-5.1 и Радуга-5.1(П) – программы для решения стационарного уравнения переноса в 2-х и 3-х мерных геометриях на одно- и многопроцессорных ЭВМ. http://www.neutronica.ru/doc/2005/Neutr_01.rar
2. Марчук Г.И., Лебедев В.И. Численные методы в теории переноса нейтронов, Москва, Атомиздат, 1981
3. Басс Л.П., Волощенко А.М., Гермогенова Т.А. Методы дискретных ординат в задачах о переносе излучения, Монография ИПМ им. М.В.Келдыша РАН, 1986.
4. Mobley C.D., Sundman L.K., Boss.E. Phase function effects on oceanic light fields. // Appl Opt 2002. V. 41. P. 1035-1050