

АТМОСФЕРНАЯ РАДИАЦИЯ, ОПТИЧЕСКАЯ ПОГОДА И КЛИМАТ

УДК 551.521.31

Оптимизация последовательного программного кода для моделирования переноса солнечного излучения в вертикально-неоднородной среде

Т.В. Русскова, Т.Б. Журавлева*

*Институт оптики атмосферы им. В.Е. Зуева СО РАН
634055, г. Томск, пл. Академика Зуева, 1*

Поступила в редакцию 13.04.2016 г.

Настоящая статья принадлежит циклу работ, направленных на повышение производительности радиационных кодов, реализующих статистический метод Монте-Карло. Приведено краткое описание основных блоков двух программ, предназначенных для расчета интенсивности солнечного излучения в вертикально-неоднородной среде, – базовой, реализованной на языке Фортран, и ее оптимизированной версии, реализованной на языке Си. Представлены результаты тестов, направленных на оценку быстродействия каждого из кодов при разных условиях численного эксперимента. В рассмотренных случаях показатели быстродействия оптимизированного Си-кода оказались выше по сравнению с показателями базового варианта. Показано, что различия во времени выполнения кодов снижаются при увеличении оптической плотности атмосферы и при использовании более производительных вычислительных машин. Разработанная Си-программа может служить основой для создания высокопроизводительного радиационного кода.

Ключевые слова: солнечная радиация, облачность, метод Монте-Карло, численное моделирование, оптимизация, языки программирования Фортран и Си; solar radiation, clouds, Monte Carlo method, numerical simulation, optimization, Fortran and C programming languages.

Введение

В атмосферной оптике имеется целый ряд задач, одним из этапов практической реализации которых является как можно более точное решение уравнения переноса солнечного излучения. Это задачи, связанные, прежде всего, с интерпретацией и обращением оптических наблюдений. С помощью измерительных приборов современного поколения, размещенных как на Земле, так и на спутниковых носителях, происходит непрерывное накопление больших объемов информации, требующей быстрой обработки для ее своевременной интерпретации. Современные тенденции в обработке данных предъявляют все более высокие требования к скорости вычислений и эффективной оптимизации расчетных программ.

В ИОА СО РАН уже на протяжении нескольких лет ведется работа по созданию алгоритмов моделирования нисходящей и восходящей солнечной радиации в сложных рассеивающих и поглощающих средах (разорванная облачность, аннотропная поверхность и др.) методом Монте-Карло. Среди численных методов решения уравнения переноса метод Монте-Карло занимает особое место: уравнение трудно разрешимо классическими методами вычислительной математики, если учитываются

реальные объекты и неоднородность среды [1–5]. Главным и, пожалуй, единственным недостатком метода являются значительные затраты машинного времени, необходимые для проведения расчетов. Потенциальным способом снижения этих затрат является создание эффективных, высокопроизводительных кодов. Учитывая специфику статистических алгоритмов, эта проблема может быть решена путем разработки параллельных версий программ, например с использованием все более популярных GPU-ускорителей (Graphics Processing Unit) [6].

Развиваемые нами алгоритмы, входящие в состав пакета MATHART (Monte CARlo Codes for THree-DimensionAl Radiative Transfer), реализованы на языке Фортран. Для создания параллельных версий целесообразно использовать их аналоги, реализованные на других языках программирования. В настоящей статье обсуждаются вопросы, связанные с конвертацией одного из алгоритмов – последовательного Фортран-кода, предназначенного для моделирования переноса солнечного излучения при наблюдении с поверхности Земли и из космоса в рамках вертикально-неоднородной плоскопараллельной модели атмосферы методом Монте-Карло – в код на языке Си и последующей его оптимизацией. Принимая во внимание современный уровень мощности вычислительных средств, нельзя не отметить, что в отличие от параллельных реализаций *существенного* повышения производительности последовательных программ (т.е. таких, операторы

* Татьяна Владимировна Русскова (btv@iao.ru);
Татьяна Борисовна Журавлева (ztb@iao.ru).

которых выполняются последовательно один за другим при одном потоке управления) за счет их оптимизации с использованием только языковых средств ожидать не стоит. Тем не менее благодаря более рациональному использованию памяти, оптимизации структуры и ряда операций последовательного кода можно добиться некоторого увеличения скорости вычислений. Кроме того, такой код может служить основой при разработке высокопроизводительных кодов для моделирования переноса солнечного излучения в более сложных средах и с использованием других алгоритмов метода Монте-Карло (локальная оценка, модифицированная двойная локальная оценка и т.д.). Результатам разработки параллельной версии кода будет посвящена следующая часть цикла работ.

1. О выборе языка программирования

Язык программирования Фортран (FORTRAN – FORMula TRANslator), появившийся в 1957 г. преимущественно для проведения научных и инженерных расчетов, долгое время оставался самым востребованным инструментом для численного моделирования в научной среде, отличаясь своими вычислительными возможностями и простотой синтаксиса. Как следствие, за более чем полувековой период его существования было создано огромное количество программ и математических библиотек, готовых к практическому использованию. В области моделирования радиационного переноса таковыми являются, например, программные комплексы РАДУГА [7], MODTRAN [8], SBDART [9] и др. Программы на новом языке высокого уровня Си, появившемся только в 1972 г., были медленнее, чем их аналоги, реализованные на Фортране. Программистам языка Си часто приходилось существенно перерабатывать код, чтобы добиться приемлемой производительности, сопоставимой с производительностью Фортран-кода. Однако в последние годы благодаря сочетанию высокооптимизированных компиляторов Си и новой методологии проектирования библиотек показатели быстродействия таких программ заметно выросли: программист может наслаждаться выразительностью и легкостью использования языка без ущерба для производительности [10]. Развивается и Фортран, однако это, в свою очередь, означает, что для реализации максимальных вычислительных и синтаксических возможностей языка требуется перевод написанных ранее многочисленных кодов на современный Фортран, влекущий за собой затраты времени и сил.

В настоящее время на языке Си также разработано целое множество не менее эффективных математических библиотек, среди которых можно отметить Intel® MKL, FFTW, CERN Program Library и др. Язык Си предоставляет широкие возможности как для системного программирования, так и для написания прикладных программ, обеспечивая гибкость и переносимость программ, логическую стройность и лаконичность записи алгоритмов, возможность разработки кодов, сопоставимых

по скорости выполнения с их ассемблер-аналогами, набор низкоуровневых средств, обеспечивающих доступ к аппаратным средствам компьютера. Прикладная программа, реализованная на языке Си, в свою очередь, может быть легко встроена в какую-либо современную вычислительную систему.

В частности, разрабатываемый авторами статьи пакет прикладных программ MATHART при их конвертации на язык Си и последующей оптимизации может быть легко встроено в высокопроизводительную систему HiPeCS (High Performance Computer System for MATHART), разработка которой ведется нами в настоящее время. Кроме того, в кодах MATHART не задействованы реализации сторонних вычислительных библиотек. Все опции и расширения добавляются в код по мере необходимости непосредственными разработчиками. Что касается ситуации в мировой практике, то большое количество программных пакетов, реализующих метод Монте-Карло и успешно применяемых в различных областях знаний – геофизике, биофизике, медицине и др., уже конвертировано на язык Си/C++ [11] либо изначально создано на его основе [12–14].

По этим причинам для реализации радиационного кода выбран язык высокого уровня Си. Предметом дальнейших наших рассуждений будет оценка быстродействия двух кодов – базового, реализованного на языке Фортран, и его оптимизированной версии, реализованной на языке Си. В качестве расчетного алгоритма метода Монте-Карло выбран алгоритм сопряженных блужданий, позволяющий вычислять интенсивность многократно рассеянного солнечного излучения при наблюдении с поверхности Земли и из космоса одновременно в нескольких заданных направлениях как в безоблачной, так и облачной горизонтально-однородной атмосфере [15].

2. Основные подходы к оптимизации

Измерение времени выполнения прикладной программы или ее отдельных блоков является одним из основных способов контроля характеристик аппаратного и программного обеспечения с точки зрения быстродействия. Такой контроль, с одной стороны, полезен для определения «узких мест» алгоритма и, с другой стороны, позволяет судить о реальной производительности компьютера. Таким образом, перед началом оптимизации любого кода прежде всего следует выявить его наиболее медленно исполняемые участки.

В основе метода Монте-Карло лежит статистический подход, позволяющий моделировать непосредственно физический процесс и статистически оценивать искомые функционалы. Идея метода состоит в моделировании случайной марковской цепи столкновений фотонов с веществом, если речь идет о переносе солнечной радиации в атмосфере Земли. На каждом шаге случайным образом определяются направление движения фотона и длина его свободного пробега. Моделирование траектории фотона завершается либо поглощением фотона средой,

либо его вылетом за ее границы. Траектории фотонов объединяются в пакеты, каждый пакет обладает определенным статистическим весом.

Весовая модификация метода, используемая в рамках данного исследования, позволяет повысить эффективность расчетов и увеличить получаемую с траектории необходимую статистическую информацию. Возникающее при этом смещение компенсируется с помощью введения специальных весовых коэффициентов [16].

Любой код, реализующий указанный метод, представляет собой совокупность логически связанных блоков, подробно описанных в [15]. К основным из них, касающимся непосредственно построения траектории, относятся блоки: моделирования длины свободного пробега фотона, определения оптического пути от точки столкновения до верхней границы атмосферы, оценки линейных функционалов, моделирования отражения излучения от подстилающей поверхности, накопления и обработки результатов.

Статистический подход к решению уравнения переноса излучения предполагает моделирование *большого* числа траекторий, а значит, многократное выполнение вышеперечисленных блоков: количество их проходов в несколько раз превышает общее количество траекторий во всех пачках. В оптически плотной среде, когда фотон претерпевает многочисленные акты рассеяния, моделирование большого количества траекторий приводит к существенным затратам машинного времени. Таким образом, наиболее медленно исполняемый участок любого кода, реализующего метод Монте-Карло, связан непосредственно с моделированием траекторий (MT) фотонов. Этому модулю программы и связанным с ним операциям и было уделено особое внимание при оптимизации кода в рамках настоящего исследования.

На рис. 1 представлена общая схема исходной Фортран-программы «MonteCarlo.for», которая включает в себя основную программу и вызываемые ею подпрограммы и функции. Основная программа содержит сегмент с объявлением переменных, сегменты чтения исходных данных, их преобразования и подготовки к дальнейшему использованию, сегмент собственно статистического моделирования и сегмент записи результатов в файл. Переменные, обращение к которым осуществляется из разных модулей (подпрограмм и функций), объединены в тематические COMMON-блоки. Все рутинные операции, такие как чтение из файлов исходных данных, их подготовка для статистического моделирования, вывод результатов расчетов в файл, выполняются в основной программе.

В результате конвертации программы на язык Си изменения коснулись как структуры исходного кода, так и отдельных его элементов. Одним из основных принципов, следование которым выполнялось при разработке новой версии программы, являлось создание четко структурированного и прозрачного кода.

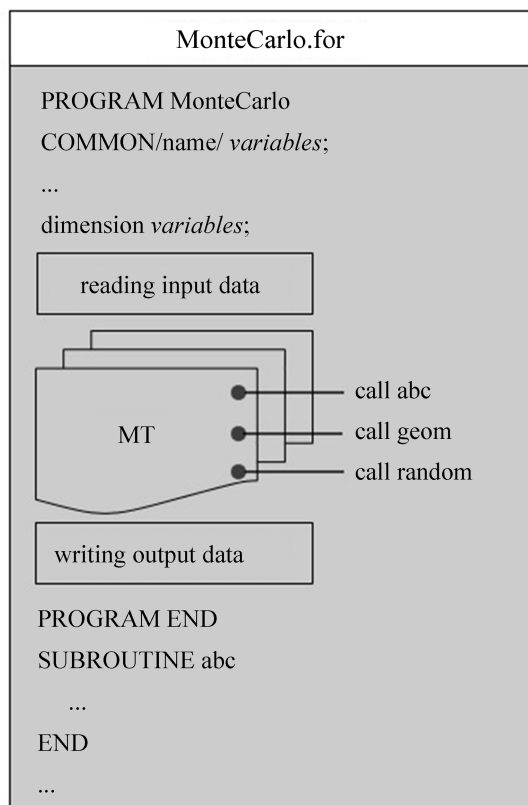


Рис. 1. Схематическое представление программного кода, реализованного на языке Фортран

Си-программа включает в себя два файла – «MonteCarlo.h» и «MonteCarlo.c» (рис. 2). Файл «MonteCarlo.h» представляет собой заголовочный файл, определяющий основные структуры данных и макросы. Главные переменные программы логически сгруппированы с использованием глобальных структур MonteCarloStruct(), ScatterStruct() и PhotonStruct(). Вместо описания функции SIGN(), обращение к которой присутствует в часто используемом фрагменте кода, введено ее макроопределение, благодаря чему программа транслируется медленнее, но выполняется быстрее.

В файле «MonteCarlo.c» определена главная функция main() и набор вспомогательных функций (void abc(PhotonStruct*) – моделирование нового направления движения фотона; void geom(PhotonStruct*, double[], double[], int) – определение расстояния от точки столкновения до верхней границы атмосферы; double random(int) – моделирование случайного числа; и др.). В отдельную функцию InputData(MonteCarloStruct*) вынесена часть кода, отвечающая за чтение из входных файлов и предварительную обработку исходных данных. Таким образом, в отличие от базовой версии в памяти программы хранится не весь набор переменных, значительная часть из которых необходима только для вывода информации пользователю, а лишь те параметры, которые задействованы непосредственно в моделировании траектории. В теле функции

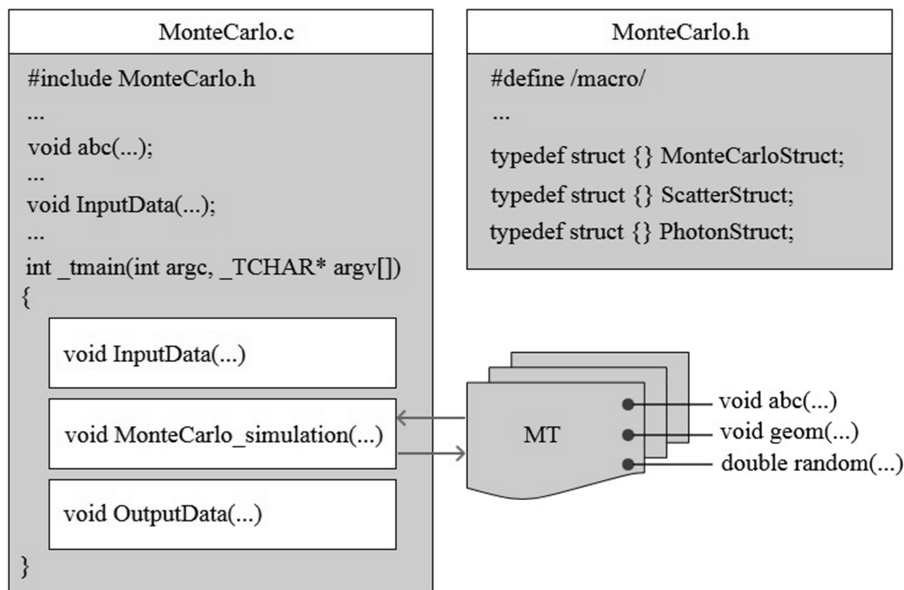


Рис. 2. Схематическое представление программного кода, реализованного на языке Си

InputData(MonteCarloStruct*) удалено также несколько переменных, дублирующих входную информацию, а для временного хранения ряда параметров использованы динамические массивы с целью более эффективного использования памяти компьютера. Аналогично в отдельные функции вынесены процедура моделирования траекторий MonteCarlo_simulation(MonteCarloStruct*) и процедура вывода результатов расчетов в файл OutputData(MonteCarloStruct*). Таким образом, программа приобрела более структурированный вид — каждая подзадача выделена в отдельный блок кода.

В блоке моделирования траекторий фотонов, а также в ряде вспомогательных функций выполнено преобразование базового кода путем замены операторов *goto*, нарушающих его читаемость и прозрачность, операторами цикла *for* и *while*. Очевидно, что чем больше операций выполняется внутри тела цикла, тем больше времени занимает цикл. При этом некоторые математические операции являются в большей степени вычислительно затратными: например, операция деления выполняется в несколько раз медленнее операции умножения. После предварительного удаления избыточных команд операция деления, как наиболее дорогая с точки зрения затрат машинного времени, заменена умножением. Например, при многократно повторяющемся расчете шага фотона до следующего акта взаимодействия представляется целесообразным вместо хранения в памяти значений коэффициента ослабления для каждого элемента среды хранить предварительно рассчитанные значения, обратные этой величине.

Вопрос о том, насколько эффективными с точки зрения быстродействия кода оказались оптимизационные процедуры, будет обсуждаться в разд. 3 и 4. Существуют и иные способы оптимизации, заключающиеся, например, в применении векторизации,

включении в код ассемблерных вставок и др. Однако в целях поддержания преемственности и сохранения читаемости программы для пользователей, не владеющих программированием профессионально, эти способы оптимизации не были использованы. Эффективность и особенности их реализации будут рассмотрены в последующих работах, направленных на разработку высокопроизводительного радиационного кода.

3. Результаты тестирования

Одной из целей настоящей работы является оценка быстродействия каждого из двух рассматриваемых кодов. Оценка выполняется для сегмента кода, связанного с моделированием траекторий фотонов, как наиболее уязвимого с точки зрения времени его выполнения. Время выполнения T (с) сегмента кода определялось разницей показаний таймера, снимаемых перед началом исполнения сегмента и после его завершения. Для каждого численного эксперимента осуществлялась серия запусков, поскольку независимо от используемого таймера измерение времени работы блока программы всегда выполняется с некоторой погрешностью. В качестве величины T анализируемого участка кода выбирался минимальный временной интервал, наблюдаемый во всей серии запусков.

В ходе тестирования оценивались следующие параметры: время выполнения T программы, ее быстродействие V (под быстродействием здесь понимается количество моделируемых пачек (n) в секунду п/с), а также относительное различие d (%), определяемое как

$$d = \left[1 - \frac{T_C}{T_F} \right] 100\%,$$

где T_C – время выполнения анализируемого блока программы на языке Си, а T_F – время выполнения блока Фортран-программы.

При тестировании использованы компиляторы Intel C++ Compiler XE 13.0 и Intel Fortran Compiler XE 13.0.0.089, встроенные в интегрированную среду Microsoft Visual Studio. В качестве измерителя времени в Си-коде использовалась функция clock(), а в Фортран-коде – функция CPU_TIME().

Численные эксперименты по расчету яркости неба при наблюдении с поверхности Земли проведены для зенитного угла Солнца, равного 70°, и зенитного угла приемника 60° в 40 азимутальных направлениях визирования в диапазоне от 0 до 180°. Предполагается плоскопараллельная, вертикально-неоднородная аэрозольно-молекулярная модель атмосферы. Поверхность Земли считается ламбертовской. Расчеты проведены вне полос поглощения излучения атмосферными газами в двух спектральных каналах λ : 0,44 и 0,87 мкм. Оптические свойства аэрозоля соответствуют типу continental average (модель OPAC) [17]. Путем вариации счетной концентрации аэрозольных частиц сформированы случаи как с повышенным замутнением атмосферы (аэрозольная оптическая толща (АОТ) не менее 0,45 на длине волны 0,44 мкм), так и высокой прозрачностью (табл. 1). Вертикальные профили температуры и атмосферного давления соответствуют летним и зимним условиям умеренных широт [18]. Альbedo подстилающей поверхности соответствует типам «смешанный лес» и «снег» [19].

Условия численных экспериментов при наличии облачности аналогичны описанным выше. Рассматриваются два типа облачности – плотная слоистая (Stratus Opacus (*StO*)) и перистая (Cirrus (*Ci*)). Оптические характеристики облаков (альbedo однократного рассеяния, индикатриса рассеяния) заимствованы из [17] (модели Stratus continental и Cirrus3). Оптическая толщина (ОТ) слоистых облаков полагалась равной 5, перистых – 2,5. Облачность моделировалась путем введения сплошного плоского слоя толщиной 1 км на соответствующих высотах – от 1 до 2 км (слоистые облака) и от 7 до 8 км (перистые облака).

Ошибка расчета яркости не превышает 0,5% в случае безоблачной атмосферы и 1% – в случае облачной. В первом случае количество моделируемых траекторий фотонов положено равным 20000 для каждой из 250 пачек (общая статистика $n = 5$ млн траекторий). В случае облачной атмосферы количество траекторий в пачке задано таким, чтобы погрешность расчета не превышала заявленную величину (40000 – плотная слоистая облачность ($n = 10$ млн), 160000 – перистая облачность ($n = 40$ млн)).

Относительные различия результатов, полученных по той и другой программе, – не более 1%. Расчеты проведены с двойной точностью и выполнены на базе самого мощного из имеющихся в распоряжении авторов персонального компьютера (ПК) – ПК I (табл. 2).

В табл. 1 приведены значения анализируемых параметров, полученные в численных экспериментах.

Таблица 1

Относительное различие d , время выполнения T и быстроедействие V блока МТ Си- и Фортран-кодов

№ эксперимента	Время года	λ , мкм	Альbedo	АОТ	Облачность		T_C , с	T_F , с	V_C , п/с	V_F , п/с	d , %
					Тип	ОТ					
1				0,15			11,84	13,14	21,1	19,0	9,9
2				0,45	–	–	22,03	23,32	11,4	10,7	5,5
3		0,44	0,0529	1,0			42,45	44,16	5,9	5,7	3,9
4	Лето			0,15	<i>Ci</i>	2,5	794,85	819,04	0,32	0,31	2,9
5				0,15	<i>StO</i>	5,0	413,0	420,9	0,6	0,6	1,9
6		0,87	0,5343	0,08	–	–	3,09	3,90	80,9	64,1	20,8
7				1,0	–	–	29,92	31,15	8,4	8,0	3,9
8	Зима			0,08	–	–	3,07	3,90	81,4	64,1	21,3
9			0,87	0,7418	0,08	<i>Ci</i>	2,5	659,3	682,8	0,38	0,36

Таблица 2

Основные характеристики ПК, используемых для проведения тестов

№ ПК	Процессор	Количество ядер	Тактовая частота, ГГц	Объем оперативной памяти, Гб	Версия ОС Windows
I	Intel Core i7-3770K	4	3,5	16	7
II	Dual Intel Pentium G860	2	3,0	3,2	XP
III	Intel Core i3-M350	2	2,27	3,0	7
IV	Intel Core i3-3217U	1	1,8	4,0	8
V	Intel Atom N270	1	1,6	1,0	7

Видно, что время выполнения оптимизированной программы на языке Си меньше времени выполнения Фортран-программы в среднем на 10%. При этом относительное различие d тем меньше, чем выше оптическая плотность атмосферы ($AOT = 1,0$, $d \approx 4\%$), и тем больше, чем среда прозрачнее ($AOT = 0,08$, $d \approx 20\%$).

Очевидно, что в облачной среде эффект оптимизации кода становится минимальным: например, величина d составляет всего $\approx 2\%$ при наличии плотной слоистой облачности. Однако при этом время работы алгоритма в той и другой языковой реализации стремительно увеличивается.

Нельзя не отметить, что каждый из выполненных численных экспериментов представляет собой единичный расчет интенсивности солнечного излучения в одном спектральном канале, для одной модели аэрозольных частиц, одного вертикального профиля температуры и давления и т.д. При решении практических задач часто приходится сталкиваться с необходимостью проведения вычислений для целых диапазонов значений этих и других параметров, например для совокупности направлений визирования при разном положении Солнца. Таким образом, создание как можно более быстрого кода, несомненно, имеет высокую практическую значимость.

4. Зависимость быстродействия кода от конфигурации ПК

Известно, что на время выполнения программы влияют разные факторы: характеристики самой программы, архитектура и конфигурация компьютера, операционная система, совместно работающие процессы, состояние компьютера на момент старта программы, компилятор и т.п. Отделить влияние интересующих характеристик от прочих далеко не всегда просто и не всегда возможно.

Рассмотрим результаты тестирования базовой и оптимизированной версий кода на компьютерах с разной конфигурацией. Для сравнительного анализа было выбрано несколько ПК под управлением операционной системы (ОС) Windows. Все из процессоров выбранных ПК принадлежат семейству Intel. Основные характеристики используемых ПК представлены в табл. 2.

На рис. 3 представлены результаты проведенного тестирования: время выполнения T анализируемого сегмента кода и относительное различие d в зависимости от конфигурации ПК. Тесты выполнены для случая 2 безоблачной атмосферы (лето умеренных широт, спектральный канал 0,44 мкм, AOT 0,45, см. табл. 1).

Как и предполагалось, время исполнения кода зависит от производительности и конфигурации компьютера. Время выполнения анализируемого сегмента кода на самом мощном ПК I примерно в 12 раз меньше времени его выполнения на наиболее слабом из представленного списка. В то же время значение T_F превышает значение T_C на тем

большую величину, чем менее производительна вычислительная машина: на ПК V величина d достигает $\approx 17\%$.

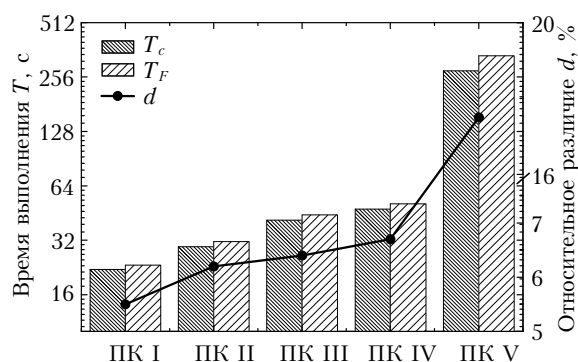


Рис. 3. Время выполнения T и относительное различие d Си- и Фортран-кодов на разных ПК для случая 2

Аналогичным образом были проведены численные эксперименты и для случаев облачной атмосферы. Результаты этих экспериментов не нарушают справедливость сформулированных выше выводов.

Заключение

Обсуждаются вопросы, связанные с конвертацией базового Фортран-кода, реализованного для моделирования переноса солнечного излучения в рамках вертикально-неоднородной плоскопараллельной модели атмосферы методом Монте-Карло, в код на языке Си и последующей его оптимизацией.

Разработан радиационный код на языке программирования высокого уровня Си. Проведен анализ эффективности кода, выявлены направления его оптимизации. Благодаря различным структурным преобразованиям, рационализации использования памяти, оптимизации циклов и другим приемам достигнута более высокая прозрачность кода, улучшена читаемость и обеспечена логическая структурированность программы без ущерба для ее быстродействия. В рассмотренных случаях оптимизированный Си-код оказался более эффективным для моделирования переноса солнечного излучения как в безоблачной (до 20%), так и облачной (до 2%) средах с точки зрения времени его выполнения по сравнению с базовым Фортран-кодом.

Показано, что различия во времени выполнения Си- и Фортран-кодов снижаются при использовании более производительных вычислительных машин и увеличении оптической плотности атмосферы, в том числе за счет появления облачности. Перспективным подходом к ускорению вычислений в облачной атмосфере является технология распараллеливания алгоритма. Разработанная Си-программа может служить основой для создания высокопроизводительного радиационного кода.

Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 16-31-60057 мол_а_дк.

1. *Boas D., Culver J., Stott J., Dunn A.* Three dimensional Monte Carlo code for photon migration through complex heterogeneous media including the adult human head // *Opt. Express*. 2002. V. 10, N 3. P. 159–170.
2. *Demers H., Poirier-Demers N., Couture A.R., Joly D., Guilmain M., Jonge N., Drouin D.* Three-Dimensional Electron Microscopy Simulation with the CASINO Monte Carlo Software // *Scanning*. 2011. V. 33, iss. 3. P. 135–146.
3. *Fang Q., Kaeli D.R.* Accelerating mesh-based Monte Carlo method on modern CPU architectures // *Biomed. Opt. Express*. 2012. V. 3, iss. 12. P. 3223–3230.
4. *Mohanty S.K., Lakshminarayanan V.* Optical Techniques in Optogenetics // *J. Mod. Opt.* 2015. V. 62, N 12. P. 949–970.
5. *3D Radiative Transfer in Cloudy Atmospheres* / Ed. by A. Marshak, A. Davis. Berlin; Heidelberg: Springer, 2005. 688 p.
6. *Owens J.D., Houston M., Luebke D., Green S., Stone J.E., Phillips J.C.* GPU computing // *Proc. IEEE*. 2008. V. 96, N 5. P. 879–899.
7. *Басс Л.П., Николаева О.В.* Радуга-6 – программа расчета стационарных и нестационарных нейтронных и гамма полей в 1D, 2D, 3D областях // Труды 7-й МНТК «Обеспечение безопасности АЭС с ВВЭР», ОКБ «ГИДРОПРЕСС», 2011.
8. *Berk A., Anderson G.P., Acharya P.K., Bernstein L.S., Muratov L., Lee J., Fox M., Adler-Golden S.M., Chetwynd J.H., Hoke M.L., Lockwood R.B., Gardner J.A., Cooley T.W., Borel C.C., Lewis P.E., Shettle E.P.* MODTRAN5: 2006 Update // *Proc. SPIE*. 2006. V. 6233, N 62331F. DOI: 10.1117/12.665077.
9. *Ricchiazzi P., Yang S., Gautier C., Sowle D.* SBDART: A Research and Teaching Software Tool for Plane-Parallel Radiative Transfer in the Earth's Atmosphere // *Bull. Amer. Meteorol. Soc.* 1998. V. 79, N 10. P. 2101–2114.
10. *Grosse-Kunstleve R.W., Terwilliger T.C., Sauter N.K., Adams P.D.* Automatic Fortran to C++ conversion with FABLE // *Source Code Biol. Med.* 2012. V. 7. Article number 5. DOI: 10.1186/1751-0473-7-5.
11. *Buras R., Dowling T., Emde C.* New secondary-scattering correction in DISORT with increased efficiency for forward scattering // *J. Quant. Spectrosc. Radiat. Transfer*. 2011. V. 112, N 12. P. 2028–2034.
12. URL: <http://reef.atmos.colostate.edu/~gregm/xrtm/>
13. *Buehler S.A., Eriksson P., Kuhn T., von Engel A., Verdes C.* ARTS, the Atmospheric Radiative Transfer Simulator // *J. Quant. Spectrosc. Radiat. Transfer*. 2005 V. 91, N 1. P. 65–93.
14. *Wang L.H., Jacques S.L., Zheng L.* MCML – Monte Carlo modeling of photon transport in multi-layered tissues // *Comp. Methods Programs Biomed.* 1995. V. 47. P. 131–146.
15. *Марчук Г.И., Михайлов Г.А., Назаралиев М.А., Дарбинян Р.А., Каргин Б.А., Елепов Б.С.* Метод Монте-Карло в атмосферной оптике. Новосибирск: Наука, 1976. 280 с.
16. *Назаралиев М.А.* Статистическое моделирование радиационных процессов в атмосфере. Новосибирск: Наука, 1990. 227 с.
17. *Hess M., Koepke P., Schult I.* Optical Properties of Aerosols and Clouds: The software package OPAC // *Bull. Amer. Meteorol. Soc.* 1998. V. 79, N 5. P. 831–844.
18. *Комаров В.С., Ломакина Н.Я.* Статистические модели пограничного слоя атмосферы Западной Сибири. Томск: Изд-во ИОА СО РАН, 2008. 222 с.
19. *Hook S.J.* ASTER Spectral Library: Johns Hopkins University (JHU) spectral library; Jet Propulsion Laboratory (JPL) spectral library; The United States Geological Survey (USGS-Reston) spectral library. 1998. Dedicated CD-ROM. Version 1.2. (см. также <http://speclib.jpl.nasa.gov>).

T.V. Russkova, T.B. Zhuravleva. The optimization of sequential code for simulation of solar radiative transfer in a vertically heterogeneous environment.

Article belongs to the series of works aimed at improving the performance of radiation codes that implement the statistical Monte Carlo method. A brief description of the main blocks of the two programs for calculating the intensity of solar radiation in a vertically inhomogeneous medium, basis, in FORTRAN, and its optimized version, in the C language, is presented. The results of tests aimed at assessing the performance of each of the code under different conditions of numerical experiment are presented. In the cases examined, the performance indicators of optimized C code were higher as compared with the basis one. It is shown that differences in execution time of the codes are reduced by increasing the optical density of the atmosphere, and using more efficient computers. Developed C program can serve as a basis for creating a high-performance radiation code.