УДК 004.942 doi: 10.15827/0236-235X.150.269-279 2025. Т. 38. № 2. С. 269–279

Увеличение производительности моделирования радиационного переноса при помощи параллельной программы Toolkit for Particle Transport 3

А.А. Галюзов [1]✉, М.В. Косов [1]

[1] Всероссийский научно-исследовательский институт автоматики им. Н.Л. Духова, г. Москва, 127030, Россия

Ссылка для цитирования
Галюзов А.А., Косов М.В. Увеличение производительности моделирования радиационного переноса при помощи параллельной программы Toolkit for Particle Transport 3 // Программные продукты и системы. 2025. Т. 38. № 2. С. 269–279. doi: 10.15827/0236-235X.150.269-279
Информация о статье
Группа специальностей ВАК: 1.2.2
Поступила в редакцию: 21.06.2024 После доработки: 03.10.2024 Принята к публикации: 22.10.2024

Аннотация. В статье описана высокопроизводительная программа Toolkit for Particle Transport 3 (TPT3), разработанная для параллельного моделирования переноса радиации. Программа эффективно использует возможности параллельных и векторизованных вычислений и работает как на центральных процессорах, так и на графических ускорителях. Это потребовало упрощения описания геометрии установок до воксельного уровня и ограничения алгоритмов взаимодействия до уровня реакций один-в-два, когда при взаимодействии частицы с веществом рождаются только две частицы. В то же время в TPT3 перенос частиц производится не независимо, а в виде многомиллионного ансамбля весовых частиц, что позволяет совмещать TPT3-расчеты с гидродинамическими алгоритмами. Продемонстрировано, что использование программы TPT3 в задачах моделирования развития ионного каскада и выхода нейтронов нейтронного генератора на многоядерных центральных процессорах обеспечивает прирост производительности в 1.5 раза по сравнению с аналогичным псевдопараллельным моделированием программы Geant4, а использование графических ускорителей позволяет повысить производительность еще на порядок. Помимо этого, в TPT3 разработан алгоритм весового моделирования переноса нейтронов, основанный на методе весовых окон, для расчета развития цепной реакции ядерного деления с характерным экспоненциальным ростом числа реальных нейтронов при сохранении ограниченного числа моделируемых весовых нейтронов. Алгоритм весового моделирования TPT3 также позволяет моделировать выгорание ядерного топлива, наработку осколков деления и поток делительных γ-квантов.

Ключевые слова: моделирование переноса радиации, CPU, GPU, стандарты препроцессорных директив OpenMP и OpenAcc, параллельные вычисления, программа TPT3, программный комплекс Geant4, программа MCNP

Введение. Начиная с середины 80-х гг. двадцатого столетия тактовая частота центральных процессоров постоянно увеличивалась. В соответствии с так называемым законом Мура частота CPU удваивалась примерно каждые 18 месяцев. Таким образом, без всяких усилий со стороны разработчиков постоянно увеличивалась производительность программных кодов физического моделирования. Но примерно с 2006 года, когда техпроцесс полупроводниковых производств стал приближаться к физическим ограничениям, обусловленным квантовым поведением микроскопических атомных систем размером в несколько нанометров, а также ограничениями тепловыделения микросхем, прирост частоты выпускаемых процессоров быстро прекратился. С тех пор из-за невозможности увеличивать тактовую частоту производители чипов стали внедрять многоядерные архитектуры. В них дополнительные возможности увеличения производительности выпускаемых вычислительных систем стали доступны только в рамках параллельного запуска расчетов на всех вычислительных ядрах процессора, а также векторизации программного кода. Тогда же было замечено, что существующие программные коды физики высоких энергий и радиационного моделирования слабо задействуют эти возможности увеличения производительности путем распараллеливания и векторизации вычислений [1]. Поэтому в связи с непрерывным ростом объемов вычислений в задачах переноса радиации назрела необходимость в обновлении ПО для того, чтобы задействовать новые ресурсы повышения производительности расчетов в эпоху многоядерных процессоров.

Как известно, запуск параллельных и независимых вычислительных задач на каждом ядре многоядерного процессора не является единственным способом ускорения параллельных вычислений. Если алгоритм моделирования допускает разбиение вычислительной задачи на независимые подзадачи меньшего размера, то использование программного интерфейса MPI позволяет проводить расчеты параллель-

но, одновременно запуская их на узлах вычислительного кластера и тем самым увеличивая производительность вычислений.

В современном программном комплексе моделирования прохождения частиц через вещество Geant4 [2] распараллеливание по вычислительным потокам организовано так, что в каждом потоке программный код исполняется последовательно, но все вычислительные потоки запускаются параллельно и независимо друг от друга. Это не приводит к существенному приросту производительности вычислений, поскольку в отдельных потоках можно запустить независимые задачи, но позволяет экономить память, занимаемую общими БД. Моделирование внутри каждого отдельно взятого потока остается последовательным, и, поскольку потоки никак не связаны между собой, невозможно синхронизировать моделируемые частицы по времени, что критически важно, например, при совмещении моделирования переноса радиации с гидродинамическими расчетами. Кроме этого, быстродействие Geant4 не может достигать теоретических пиковых значений из-за интенсивного использования наследования и виртуальных функций, а также многоуровневой иерархии классов языка программирования C++. Еще одним фактором замедления является сложное описание моделируемых геометрических объемов, что приводит к снижению скорости счета задач со сложной геометрической структурой.

Помимо Geant4, широко распространен программный код MCNP [3–5], основанный на применении метода Монте-Карло и написанный на языке Fortran. В 2003 году при создании версии MCNP5 он был переработан, в нем была реализована поддержка распараллеливания вычислений при помощи программного интерфейса MPI и стандарта препроцессорных директив OpenMP, что позволяло ускорять расчеты, используя многоядерные процессоры и многопроцессорные вычислительные кластеры. MCNP является основной программой моделирования каротажных измерений [4, 5]. Программа MCNP хорошо зарекомендовала себя в области физики ядерных реакторов [6], в моделировании радиационной защиты [7] и эффектов радиационной медицины [8]. Ее недостатком является то, что она подпадает под экспортные ограничения министерства энергетики США и в настоящее время в РФ невозможно официально получить не только ее исходный код, но и скомпилированный исполняемый файл программы. Соответственно, ни в одной научной работе нельзя ссылаться на расчеты, проведенные при использовании этой программы.

Давно применяется и программа для моделирования переноса радиации SRIM [9]. В 1980-е гг. SRIM создавалась для моделирования задач ионной имплантации в полупроводниковых материалах, а сейчас является одной из востребованных программ в радиационном материаловедении. Ежегодно публикуется более 700 научных работ, расчеты которых получены при ее помощи. SRIM предоставляет широкую функциональность и удобный пользовательский интерфейс, что позволяет в кратчайшие сроки научиться запускать расчеты в этой программе. Однако существует проблема в том, что приложение распространяется в виде скомпилированного файла, а его исходные коды закрыты. По этой причине при усложнении моделируемой задачи возникают непреодолимые сложности, хотя программа очень удобна в использовании и позволяет быстро моделировать типовые задачи переноса радиации.

Известны отечественные разработки подобных программ. Так, программный комплекс ГЕФЕСТ позволяет определять нейтронно-физические характеристики реактора на быстрых нейтронах [10]. Он может проводить нестационарные нейтронно-физические расчеты реактора с учетом выгорания ядерного топлива.

Дополнительными стимулами к разработке независимой от стороннего ПО параллельной программы стали неослабевающий интерес к GPGPU (*General Purpose Computing on GPU*) – использованию графических ускорителей для увеличения производительности массивно-параллельных вычислений, а также бурное развитие технологий и стремительное увеличение мощности видеокарт [11–13]. Программный код, в котором присутствуют необходимый для векторизации вычислений параллелизм данных и возможность создания множества независимых параллельных вычислительных потоков, идеально соответствует архитектуре графических ускорителей, состоящих из тысяч существенно более слабых, чем на центральных процессорах, вычислительных ядер с общей памятью. Использование подхода GPGPU не позволило в разы ускорить различные научные программные коды, поскольку подавляющее большинство программ, используемых для моделирования переноса радиации (Geant4, SRIM, MCNP и др.), из-за своей сложности рассчи-

таны только на использование CPU и их нельзя портировать на графические ускорители.

Для устранения недостатков существующих программ радиационного моделирования во ВНИИА им. Н.Л. Духова (г. Москва) разработана программа параллельного моделирования переноса радиации *Toolkit for Particle Transport* (TPT3) (Свидетельство о госрегистрации в Роспатенте № 2024614877).

Программа TPT3 является развитием алгоритмов CHIPS-TPT [14, 15] до уровня независимой от Geant4 программы, интенсивно использующей распараллеливание расчетов по вычислительным потокам и векторизацию вычислений. Основные алгоритмы были перенесены из библиотеки CHIPS-TPT в программу TPT3 с учетом отличия архитектур программных кодов, распараллеливания по вычислительным потокам и оптимизации программных алгоритмов.

Со стороны разработчиков популярных в области моделирования радиационного переноса программных решений также предпринимались попытки оптимизации существующего ПО с целью повышения эффективности использования многоядерных процессоров, а также графических ускорителей для увеличения производительности вычислений. В результате сотрудничества CERN и Intel разработана параллельная векторизованная программа GeantV [16], реализующая физические алгоритмы программы Geant4, в которую была внедрена высокопроизводительная векторизованная геометрическая библиотека VecGeom [17, 18]. Однако к настоящему времени проект GeantV закрыт, поскольку переписать огромный объем последовательного кода программы Geant4 в виде, пригодном для параллельных вычислений, не представляется возможным из-за огромных трудозатрат, а также необходимости изменения и оптимизации многих алгоритмов для их корректного функционирования в параллельном режиме работы. В рамках проекта Geant4 для иллюстрации возможностей ускорения параллельных вычислений при использовании GPU в качестве примера была создана узконаправленная программа моделирования распространения оптических фотонов в детекторах частиц (в первую очередь в детекторах нейтрино) Chroma [19], работающая на GPU и написанная на языке CUDA, а также имеющая интерфейс на языке Python. Как утверждается в [20], при ее использовании удалось достичь увеличения производительности моделирования аналогичных задач от 50 до 400 раз на GPU по сравнению с Geant4 на CPU. Однако данная программа имеет узкую применимость и для моделирования большинства задач радиационного переноса не используется. Помимо этого, предпринимались попытки ускорить собственно сам программный код Geant4 [21, 22] на графических ускорителях. Также созданы различные независимые параллельные программные коды [23, 24], целью которых было ускорение алгоритмов переноса радиации путем эффективного распараллеливания вычислений. Однако из-за сложностей распараллеливания алгоритмов радиационного переноса ни одна из этих попыток пока не привела к появлению универсального высокопроизводительного параллельного программного кода, моделирующего перенос радиации.

В MCNP распараллеливание вычислений, как и в Geant4, реализовано на уровне событий, таким образом, каждая частица независимо моделируется в своем вычислительном потоке, что приводит к сложности синхронизации частиц по времени. Также код MCNP не поддерживает запуск на GPU. Портирование программы на графические ускорители не планируется, поскольку это требует очень больших усилий (https://mcnp.lanl.gov/faq.html).

Целью разработки программы TPT3 была попытка создать эффективный высокопроизводительный параллельный программный код для моделирования прикладных задач радиационной стойкости, в частности, для моделирования нейтронных генераторов, которая работала бы как на центральных процессорах, так и на графических ускорителях. В TPT3 все частицы моделируются независимо, параллельно и одновременно, поэтому элементарно синхронизировать их по времени без необходимости раскручивания стека частиц в каждом потоке, как в Geant4. Используемые алгоритмы физического моделирования в TPT3 прошли предварительную оптимизацию производительности для эффективной работы в параллельном режиме. Также в TPT3 используется воксельное описание геометрии без замедляющей расчеты сложной иерархии геометрических объемов, характерной для Geant4 и MCNP. Увеличение точности описания нетривиальных геометрических объемов достигается увеличением числа вокселей в расчетной сетке, осуществляемым простым расширением используемой оперативной памяти. При этом замедление вычислений сопоставимо с соответствующим ограничением шага моделирования в программе Geant4. Использование воксельной гео-

метрии и одновременное параллельное моделирование всех частиц позволяют синхронизировать распространение частиц по времени, а также делает простым и естественным совмещение моделирования радиационного переноса и, например, гидродинамического моделирования, что при использовании Geant4 оказывается практически невыполнимой задачей.

Алгоритм навигации моделируемых частиц в программе ТРТ3

Рабочий цикл ТРТ3-моделирования подразделяется на четыре функциональные части: пропагатор, компрессор, реактор и инжектор (рис. 1). Любая частица (фотон, электрон, позитрон, ион или нейтрон) представляется структурой из шестнадцати 32-битных полей. Такой унифицированный подход к хранению данных в программе моделирования позволяет обрабатывать массив частиц с помощью векторизованной арифметики AVX512 (AVX2). Для оптимальной организации интенсивного взаимодействия с оперативной памятью компьютера рабочий массив частиц заказывается в статической памяти. Там же располагаются вспомогательные массивы для быстрой сортировки частиц по типу взаимодействия, поскольку для повышения эффективности параллелизма данные однородных процессов должны быть локализованы в памяти.

В пропагаторе параллельно обрабатываются частицы массива, которые продвигаются до следующей вершины ядерного или дискретного электромагнитного взаимодействия (тип ir = 2, 3) или до границы вокселя (тип ir = 1), где ir – индекс типа реакции, а воксель – ячейка декартовой сетки. Если энергия частицы оказывается меньше пороговой (задается пользователем) или энергии, теряемой частицей вследствие энергетических потерь на расчетном шаге моделирования, она объявляется остановившейся (ir = 0) и удаляется из массива моделируемых частиц в компрессоре. В противном случае частица достигает следующей вершины своего трека. При этом заряженная частица теряет энергию, пропорциональную длине шага, за счет электронных энергетических потерь. Таким образом, в пропагаторе выполняется следующее:

– рассчитывается длина пробега частицы до различных вершин взаимодействия с веществом или минимальное расстояние до пересечения трека границей вокселя;

– из рассчитанных длин до взаимодействия или пересечения границ выбирается минимальная (ΔL), определяется соответствующий выигравший процесс (ir) новой вершины трека частицы;

– частица перемещается в новую вершину взаимодействия, то есть ее координата $\vec{r}(x, y, z)$ изменяется по закону $\vec{r}'(x, y, z) = \vec{r}(x, y, z) + \Delta l \cdot \vec{v}$, где $\vec{v}$ – единичный вектор в направлении скорости частицы;

– если выиграл процесс переноса до границы с соседним вокселем (ir = 1), то модифицируются индексы вокселя, в котором находится частица (*ix, jy, kz*), и на следующем шаге используются параметры среды уже соседнего вокселя;

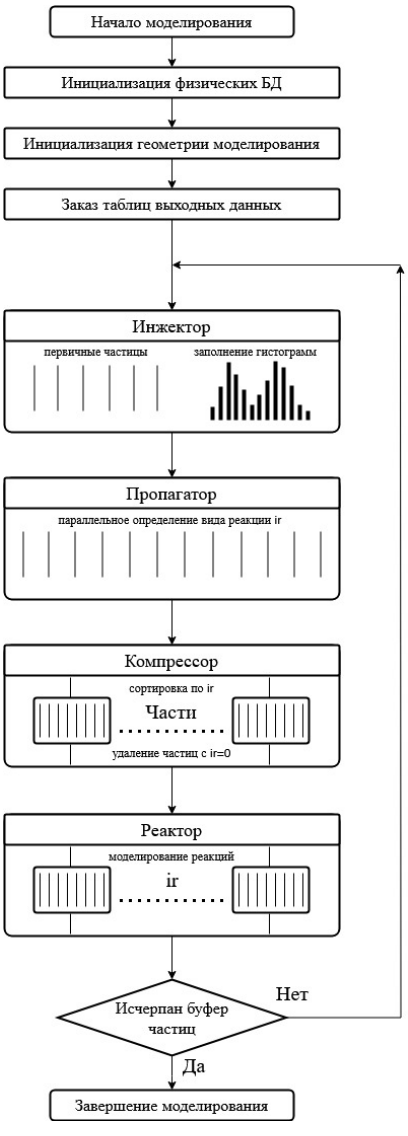

Рис. 1. Схематическое описание работы программы ТРТ3

Fig. 1. Schematic description of TPT3 operation

– на выигравшей длине ΔL для заряженных частиц рассчитываются потери энергии $\Delta E = dE/dx \cdot \Delta L$ с учетом их флуктуаций [25] (нейтральные частицы, например, нейтроны, потерь энергии на шаге за счет dE/dx не имеют);

– изменяется собственное время частицы как $t' = t + \Delta L / v$, где $v = p/(m+E)$ – безразмерная скорость частицы в скоростях света (в ТРТ3 используется принятая в ядерной физике система единиц, где скорость света c = 1, поэтому импульс и энергия измеряются в МэВ, а время в единицах расстояния); если собственное время частицы становится больше текущей временной границы (равной сумме всех предыдущих временных шагов + время текущего шага моделирования), частица останавливается в положении, при котором ее собственное время равно текущей временной границе, до начала нового временного шага моделирования;

– если потери энергии на шаге оказываются больше текущей кинетической энергии частицы E, то частица считается остановившейся (ir = 0), вся ее кинетическая энергия записывается в потерянную энергию ΔE, а сама частица после считывания потерянной энергии исключается в компрессоре из массива моделируемых частиц.

В результате работы пропагатора в рабочем массиве моделируемых частиц оказываются частицы четырех типов:

– ir = –1: остановившиеся на текущей временной границе частицы, которые должны быть скопированы в компрессоре во вспомогательный массив, где будут ожидать начала следующего временного шага;

– ir = 0: исключенные из моделирования остановившиеся либо вышедшие за границы моделируемого объема частицы, которые должны быть удалены из массива частиц в компрессоре после учета выделенной ими энергии;

– ir = 1: частицы, остановившиеся на границе вокселя и не создающие новых вторичных частиц, которые должны быть добавлены в реакторе, а потому реактор частицы с ir = 1 не обрабатывает;

– ir = 2: упруго рассеявшиеся на данном шаге частицы, а в общем случае это может быть рассеяние с неупругой модификацией существующей частицы и с добавлением новой частицы в массив частиц в результате рассеяния, например, ядра отдачи;

– ir = 3: в задаче моделирования реакции ядерного деления – частицы, принявшие участие в реакции деления $^{235}_{92}U$ нейтронами, при которой поглощенный нейтрон исчезает, а на его месте появляются весовой нейтрон, вес которого определяется спектром и множественностью всех делительных нейтронов, и весовой γ-квант, вес которого определяется спектром и множественностью всех делительных γ-квантов.

Далее в работу вступает компрессор, отвечающий за сортировку частиц по типу взаимодействия ir (рис. 2), где для краткости не показаны частицы с ir = 3. Исходный массив частиц делится на сортируемые параллельно подмассивы. Затем частицы с одинаковым значением ir (2 либо 3) в каждом из отсортированных подмассивов с соответствующими отступами копируются во вспомогательный массив для промежуточного хранения частиц с этим ir, а потом в порядке убывания ir из этих промежуточных – в исходный массив частиц. При этом частицы с ir = 0 не копируются и, таким образом, происходит их удаление. Сортировка по ir нужна для того, чтобы сначала в массиве моделируемых частиц лежали те из них, которые должны обрабатываться реактором (все частицы с ir = 3, которые исчезают и рождают две новые частицы, а потом – все с ir = 2, которые лишь модифицируются с рождением одной новой частицы), а затем перешедшие в другой воксель без взаимодействия (ir = 1), которые реактор во-

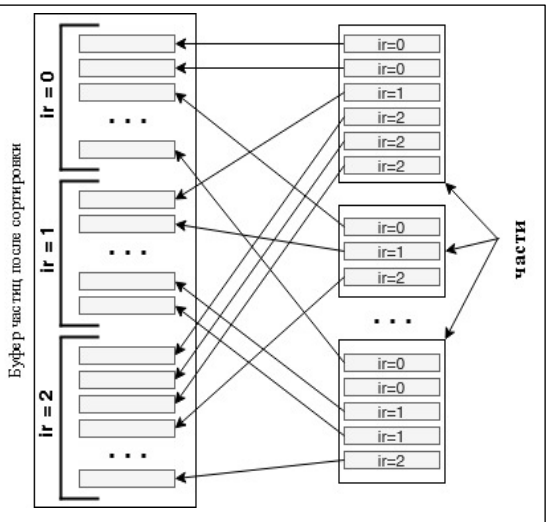

Рис. 2. Схема параллельной сортировки частиц в массиве по типу взаимодействия

Fig. 2. Scheme of parallel sorting of particles in a particle array by interaction type

обще не обрабатывает. Ожидающие следующего временного шага частицы с ir = –1 копируются в специально предназначенный для этого массив. В начале следующего временного шага, когда число моделируемых частиц исчерпывается, поскольку все они, дойдя до временной границы, оказываются перемещенными в массив для ir = –1, все частицы из этого вспомогательного массива перемещаются обратно в начало опустевшего в конце временного шага массива переносимых частиц. После исключения остановившихся частиц с ir = 0 и ждущих следующего временного шага с ir = –1 компрессор удваивает ту часть массива частиц, которая занята частицами с ir = 3, подготавливая место для новых частиц (в случае ir = 3 первая новая частица записывается на место исчезнувшей родительской). Аналогично для ir = 2 выделяется место для новой вторичной частицы (рис. 3). Таким образом, компрессор и уменьшает, и увеличивает полное число частиц в массиве переносимых частиц: $N' = N - N_0 - N_{-1} + N_2 + N_3$, где N – полное число частиц в начале шага, а N_k – число частиц с ir = k.

Если число моделируемых частиц падает ниже оптимального для интенсивных параллельных вычислений уровня, то при следующем входе в инжектор добавляются новые первичные частицы или расщепляются частицы с большими весами на тождественные частицы с меньшими весами до тех пор, пока число моделируемых частиц не достигнет этого уровня.

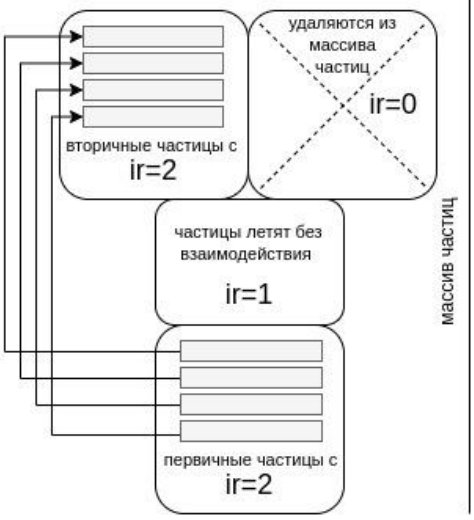

Рис. 3. Удаление частиц с ir = 0 и создание для частиц с ir = 2 и ir = 3 места для вторичных частиц

Fig. 3. Removing particles with ir = 0 and creating a place for secondary particles for particles with ir = 2 and ir = 3

Между компрессором и инжектором работает реактор, который параллельно обрабатывает нижнюю часть массива частиц, соответствующую частицам с ir = 3 и ir = 2:

– в соответствии с алгоритмом реализации конкретного физического процесса случайным образом разыгрываются импульсы $p'_{1,2}$ и энергии $E'_{1,2}$ двух вторичных частиц с сохранением энергии $E = E'_1 + E'_2$, а недостающий импульс передается веществу, поэтому его сохранение не проверяется;

– координаты родительской частицы $r(x, y, z)$ и соответствующий индекс текущего вокселя (ix, jy, kz) дублируются в новую частицу;

– устанавливается уникальный для каждого вида частиц 10-значный PDG-код [26] и определяется масса новой частицы, в случае ir = 3 эти параметры могут меняться и для родительской частицы;

– тип взаимодействия всех частиц на выходе из реактора может оставаться неопределенным, поскольку следующие за реактором инжектор и пропагатор нового шага моделирования не принимают это поле во внимание, а в пропагаторе ему все равно устанавливается новое значение.

В результате работы реактора в массиве переносимых частиц все обработанные и созданные частицы имеют неопределенный тип взаимодействия. Необработанные реактором частицы с ir = 1 сохраняют свой тип, но ни в последующем инжекторе, ни при новом вызове пропагатора этот тип не анализируется, хотя может быть использован для подготовки выходной информации и заполнения соответствующих гистограмм.

Заканчивает цикл параллельного TPT3-навигатора переносимых частиц инжектор, в котором для накопления заданной статистики добавляются первичные частицы, а также поддерживается оптимальное для высокопроизводительных параллельных расчетов число моделируемых частиц.

Сравнение производительности программы TPT3 и программного комплекса Geant4

Моделирование ионного каскада и итоговый спектр ионов кремния, замедлившихся до

энергии менее 30 кэВ, показаны на рисунке 4. В процессе моделирования условно бесконечная кубическая кремниевая мишень с длиной ребра 1 км изотропно облучалась из ее центра нейтронами с энергией 5 МэВ. При этом на CPU запускалось 10^5, а на GPU – 10^7 нейтронов. Размеры мишени были выбраны из тех соображений, что при произвольной энергии первичных нейтронов все они остановились бы в мишени и при этом ни один не вылетел бы за ее пределы. Как видно из рисунка, при нормировке на число начальных нейтронов оптимизированные алгоритмы программы TPT3 обеспечивают меньший статистический разброс, чем Geant4.

На рисунке 5 показано соотношение приведенных времен счета на один первичный нейтрон задачи моделирования ионного каскада при облучении кремниевой мишени нейтронами с энергией 5 МэВ с помощью Geant4 и TPT3 на 4-ядерном CPU Intel Core i7-3770 и на 64-ядерном CPU Intel Xeon Phi KNL-7210, а также GPU NVIDIA Titan V и новейшем графическом ускорителе NVIDIA A100 в зависимости от числа запущенных первичных частиц. Как видим, Geant4 на 4-ядерном CPU (позволяющем создавать два логических потока на каждом физическом ядре) показывает наихудшую производительность, ускорение при увеличении числа первичных частиц для него не наблюдается, поскольку это принципиально последовательный код, в котором все вычисления просто по очереди запускаются на независимых ядрах CPU. При малом числе параллельно обрабатываемых событий (10^5 первичных частиц) TPT3 незначительно отличается по производительности от последовательного Geant4, но с ростом числа частиц, то есть уже с нескольких миллионов частиц, производительность увеличивается примерно в 1.4 раза, что доказывает преимущество параллельных вычислений с использованием векторизации.

При запуске TPT3 на мультиядерном CPU Intel KNL имеется более сильная зависимость от числа первичных частиц и, начиная с $3 \cdot 10^6$ запущенных нейтронов, вычисления на KNL выходят на эффективный параллельный режим работы, превосходя Geant4 на Core i7 в 2.3 раза. При этом число вычислительных ядер и их тактовая частота соотносятся как 1.3 ГГц у 64 ядер KNL против 3.4 ГГц у 4 ядер Core i7, и прямой связи производительности и характеристик CPU не наблюдается. При этом TPT3 на CPU KNL превосходит по производительности TPT3 на Core i7 в 1.6 раза (вместо 6 раз), что свидетельствует об упрощенной архитектуре ядер в KNL.

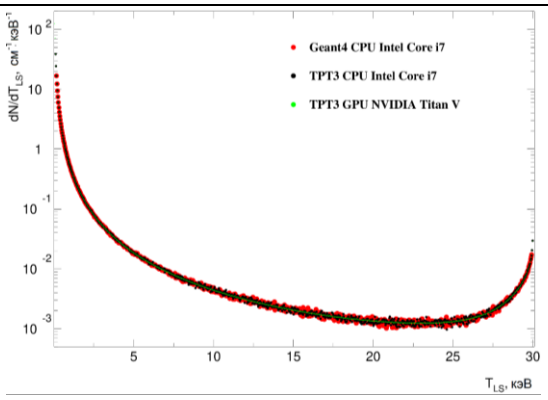

Рис. 4. Моделирование ионных каскадов, инициированных в кремниевой мишени нейтронами с начальной кинетической энергией 5 МэВ (показан спектр ионов кремния, замедлившихся до энергии менее 30 кэВ)

Fig. 4. Simulation of ion cascades initiated in a silicon target by neutrons with an initial kinetic energy of 5 MeV (the silicon ion spectrum slowed down to an energy of less than 30 keV)

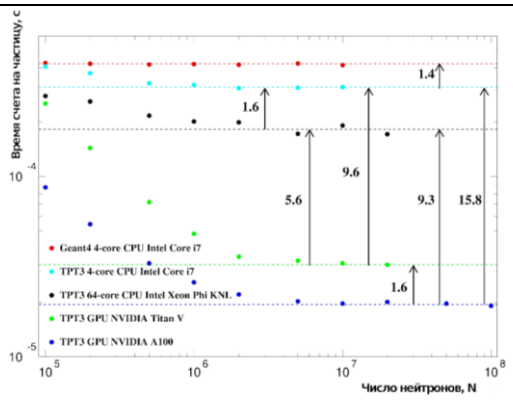

Рис. 5. Сравнение времени счета на один первичный нейтрон при моделировании ионного каскада программами TPT3 и Geant4 в зависимости от числа N запущенных нейтронов для различных вычислительных платформ

Fig. 5. Comparing the counting time for one primary neutron when simulating an ion cascade by TPT3 and Geant4 depending on the number N of launched neutrons for different computing platforms

В случае GPU прирост производительности прекращается примерно с $3 \cdot 10^6$ запущенных нейтронов, то есть при тех же значениях, что и для KNL. В эффективном параллельном режиме работы TPT3 на GPU NVIDIA Titan V превосходит Geant4 на CPU Intel Core i7 в 13.1 раза, а TPT3 на Core i7 примерно на порядок. Соответственно, TPT3 на GPU NVIDIA A100 превосходит Geant4 на Core i7 в 21 раз, а TPT3 на этом же CPU – примерно в 15 раз. При этом TPT3 на GPU Titan V в 5.6, а на GPU A100 – в 9.3 раза производительнее TPT3 на CPU KNL, а графический ускоритель A100 примерно в 1.6 раза быстрее видеокарты Titan V.

Программа TPT3 позволяет проводить весовое моделирование цепной реакции в актинидах. Проверка функциональности весового моделирования цепной реакции проводилась на урановом кубе с длиной ребра a, разделенном на 32×32×32 вокселя. При TPT3-моделировании цепной реакции в урановом кубе вторичными весовыми частицами акта деления являются один нейтрон и один γ-квант с весами, определяемыми их делительной множественностью и их спектром. Увеличение числа реальных моделируемых нейтронов не приводит к росту числа весовых нейтронов, поэтому TPT3-моделирование позволяет проводить расчеты даже при периодических граничных условиях ($a \to \infty$). На рисунке 6 показана рассчитанная зависимость отношения интенсивности потока реальных γ-квантов к интенсивности потока реальных нейтронов от длины ребра уранового куба. Пунктирная асимптота $R = 5.52$ соответствует периодическим граничным условиям ($a \to \infty$). Слева от вертикальной линии, соответствующей критической длине ребра уранового куба $a = 16.855 \pm 0.005$ см, показаны отношения для подкритических, а справа – для сверхкритических размеров куба.

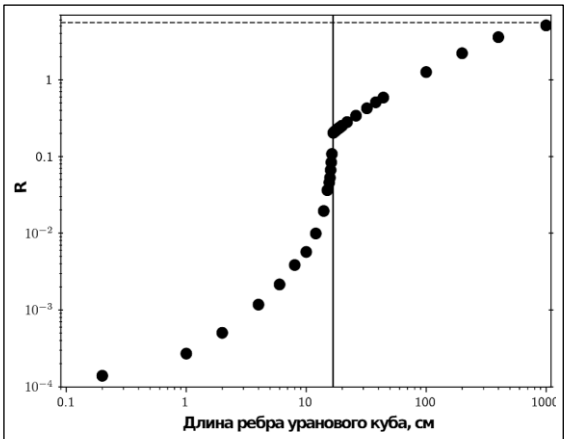

Рис. 6. Зависимость отношения интенсивностей потоков γ-квантов и нейтронов в урановом кубе от длины его ребра

Fig. 6. Dependence of the ratio of the of γ-quanta and neutron flux intensities in a uranium cube on its edge length

Заключение

Разработана кроссплатформенная высокопроизводительная программа моделирования переноса радиации TPT3, которая эффективно использует аппаратные возможности распараллеливания и векторизации программного кода на современных вычислительных устройствах и работает как на центральных процессорах (CPU), так и на графических ускорителях (GPU). Программа TPT3 может применяться для моделирования прохождения радиации через конструкционные элементы ядерных установок, и для ускорения расчетов в ней используется упрощенная воксельная геометрия.

Апробация программы TPT3 показала хорошее совпадение полученных результатов с результатами аналогичного моделирования при помощи Geant4, а также существенный прирост производительности моделирования TPT3 на GPU по сравнению с Geant4, работающим только на CPU.

Реализованная в программе TPT3 возможность синхронизации моделируемых частиц по времени открывает перспективы эффективной интеграции моделирования переноса радиации с другими программами, например, с программами гидродинамического моделирования. При совмещении с гидродинамическим моделированием, требующим расчета большой выделяемой энергии, весовое TPT3-моделирование переноса нейтронов позволяет проводить расчеты для интенсивных нейтронных потоков с учетом выгорания ядерного топлива.

Список литературы

1. Jarp S., Lazzaro A., Nowak A. The future of commodity computing and many-core versus the interests of HEP software. JPCS, 2012, vol. 396, art. 052058. doi: 10.1088/1742-6596/396/5/052058.

2. Allison J., Amako K., Apostolakis J., Arce P. et al. Recent developments in Geant4. Nucl. Instr. Methods A, 2016, vol. 835, pp. 186–225. doi: 10.1016/j.nima.2016.06.125.

3. Rising M.E., Armstrong J.C., Bolding S.R. et al. MCNP code version 6.3.0 release notes. Tech. Report LA-UR-22-33103, 2023. URL: https://mcnp.lanl.gov/pdf_files/TechReport_2023_LANL_LA-UR-22-33103Rev.1_RisingArmstrongEtAl.pdf (дата обращения: 8.08.2024). doi: 10.2172/1909545.

4. Varignier G., Fondement V., Carasco C. et al. Comparison between GEANT4 and MCNP for well logging applications. EPJ Web of Conf., 2023, vol. 288, art. 01002. doi: 10.1051/epjconf/202328801002.

5. Marchais T., Perot B., Allinei P. et al. Experimental validation of a CeBr3 gamma-ray logging probe MCNP model. EPJ Web of Conf., 2023, vol. 288, art. 05003. doi: 10.1051/epjconf/202328805003.

6. Cybowska J. Reactor power measurement using N16 and N17 production rate in MARIA reactor. EPJ Web of Conf., 2023, vol. 288, no. 04022. doi: 10.1051/epjconf/202328804022.

7. Nabil I.M., El-Samrah M.G., Omar A., Tawfic A.F., El Sayed A.F. Experimental, analytical, and simulation studies of modified concrete mix for radiation shielding in a mixed radiation field. Sci. Reports, 2023, vol. 13, art. 17637. doi: 10.1038/s41598-023-44978-8.

8. Tseng W., Furutani K., Beltran C., Lu B. An automation of Monte Carlo workflow for dosimetry study of an Elekta LINAC delivery system in radiotherapy. TipsRO, 2024, vol. 31, art. 100257. doi: 10.1016/j.tipsro.2024.100257.

9. Ziegler J., Ziegler M.D., Biersack J. The stopping and range of ions in matter. Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms, 2010, vol. 268, no. 11-12, pp. 1818–1823. doi: 10.1016/j.nimb.2010.02.091.

10. Wang Y., Zeng X., Decker J. A GPU-accelerated radiation transfer model using the lattice boltzmann method. Atmosphere, 2021, vol. 12, no. 10, art. 1316. doi: 10.3390/atmos12101316.

11. Askar T., Yergaliyev A., Shukirgaliyev B., Abdikamalov E. Exploring numba and CuPy for GPU-accelerated Monte-Carlo radiation transport. Computation, 2024, vol. 12, no. 3, art. 61. doi: 10.3390/computation12030061.

12. Silvestri S., Pecnik R. A fast GPU Monte Carlo radiative heat transfer implementation for coupling with direct numerical simulation. J. of Computational Phys. X, 2019, vol. 3, art. 100032. doi: 10.1016/j.jcpx.2019.100032.

13. Косов М.В., Савин Д.И. CHIPS-TPT. Свид. о регистр. ПрЭВМ № 2014611928. Рос. Федерация, 2014.

14. Savin D., Kosov V. Exclusive data-based modeling of neutron-nuclear reactions below 20 MeV. EPJ Web of Conf., 2017, vol. 146, art. 12028. doi: 10.1051/epjconf/201714612028.

15. Thulliez L., Jouanne C., Dumonteil E. Improvement of Geant4 Neutron-HP package: From methodology to evaluated nuclear data library. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2021, vol. 1027, art. 166187. doi: 10.1016/j.nima.2021.166187.

16. Amadio G., Apostolakis J., Bandieramonte M. et al. The GeantV project: Preparing the future of simulation, JPCS, 2015, vol. 664, art. 072006. doi: 10.1088/1742-6596/664/7/072006.

17. Amadio G., Apostolakis J., Bandieramonte M. et al. Towards a high performance geometry library for particle-detector simulations, JPCS, 2015, vol. 608, art. 012023. doi: 10.1088/1742-6596/608/1/012023.

18. Amadio G., Apostolakis J., Cosmo G., Dosaru D. et al. Offloading electromagnetic shower transport to GPUs. JPCS, 2023, vol. 2438, no. 1, art. 012055. doi: 10.1088/1742-6596/2438/1/012055.

19. Seibert S., Latorre A. Fast optical Monte-Carlo simulation with surface-based geometries using Chroma. 2011. URL: https://www.snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF2_CompF0-NF10_NF0_Chroma-045.pdf (дата обращения: 08.08.2024).

20. Kaptanoglu T., Luo M., Land B. et. al. Spectral photon sorting for large-scale Cherenkov and scintillation detectors. Phys. Review D, 2020, vol. 101, art. 072002. doi: 10.1103/PhysRevD.101.072002.

21. Okada S., Mirakami K., Incerti S., Amako K., Sasaki T. MPEXS-DNA, a new GPU-based Monte Carlo simulator for track structures and radiation chemistry at subcellurar scale. Med. Phys., 2019, vol. 46, no. 3, pp. 1483–1500. doi: 10.1002/mp.13370.

22. Valassi A., Yazgan E., McFayden J. et al. Challenges in Monte Carlo event generator software for high-luminosity LHC. Computing and Software for Big Sci., 2021, vol. 5, art. 12. doi: 10.1007/s41781-021-00055-1.

23. Li Y., Zhang B., Yang S., Chen Y. A data-driven method for calculating neutron flux distribution based on deep learning and the discrete ordinates method. Energies, 2024, vol. 17, no. 14, art. 3440. doi: 10.3390/en17143440.

24. Gimenez-Alventosa V., Gomez V.G., Oliver S. PenRed: An extensible and parallel Monte-Carlo framework for radiation transport based on PENELOPE. Computer Physics Communications, 2021, vol. 267, art. 108065. doi: 10.1016/j.cpc.2021.108065.

25. Selau F.F., Trombini H., Fadanelli R.C., Vos M., Grande P.L. On the energy-loss straggling of protons in elemental solids: The importance of electron bunching. Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms, 2021, vol. 497, pp. 70–77. doi: 10.1016/j.nimb.2021.03.006.

26. Zyla P., Barnett R., Beringer J. et al. Review of particle physics. PTEP, 2020, vol. 2020, no. 8, art. 083CO1. doi: 10.1093/ptep/ptaa104.

Software & Systems doi: 10.15827/0236-235X.150.269-279 2025, 38(2), pp. 269–279

Increasing performance of the radiation transport simulation: TPT3 parallel program

Andrey A. Galyuzov [1]✉, Mikhail V. Kosov [1]

[1] Dukhov Automatics Research Institute (VNIIA), Moscow, 127030, Russian Federation

For citation

Galyuzov, A.A., Kosov, M.V. (2025) ‘Increasing performance of the radiation transport simulation: TPT3 parallel program’, *Software & Systems*, 38(2), pp. 269–279 (in Russ.). doi: 10.15827/0236-235X.150.269-279

Article info
Received: 21.06.2024 After revision: 03.10.2024 Accepted: 22.10.2024

Abstract. The problem of increasing performance of radiation transport modeling programs is relevant for detailed calculations of radiation effect on nuclear facility structural elements. One of the ways to increase computational performance is to use multicore CPUs with vector instructions. Programs with sequential code, such as MCNP and Geant4, are able to parallelize only events initiated by independent particles, their transport simulation is performed sequentially, which does not allow full using of multicore CPU capabilities. The high-performance program TPT3 for parallel simulation of a radiation transport has been developed in the Dukhov's Institute VNIIA. It utilizes parallel computations and vectorization capabilities as efficient as possible and running on both central processors and graphics accelerators. This required simplifying the description of the installation geometry to the voxel level and restricting the interaction algorithms to one-in-two reactions, where only two particles are born when a particle interacts with matter. On the other hand, particle transport is not performed independently in TPT3, but as a multi-million weight particle ensemble, which allows combining TPT3 calculations with hydrodynamic algorithms. The paper demonstrates that using TPT3 program in the simulation tasks of an ion cascade and neutron yield in a neutron generator on multicore central processors provides a 1.5-fold performance increase compared to similar pseudo-parallel simulation of the Geant4 program. Graphics accelerators allow increasing the performance by another order of magnitude. In addition, a weighted neutron transport modeling algorithm has been developed in the TPT3 program, which makes it possible to simulate the nuclear fission chain with a characteristic exponential increase in the number of real neutrons while maintaining a limited number of simulated weighted neutrons.
Keywords: radiation transport simulation, CPU, GPU, preprocessor directive standards OpenMP and OpenAcc, parallel computations, TPT3 program, Geant4 programming package, MCNP program

References

1. Jarp, S., Lazzaro, A., Nowak, A. (2012) 'The future of commodity computing and many-core versus the interests of HEP software', *JPCS*, 396, art. 052058. doi: 10.1088/1742-6596/396/5/052058.

2. Allison, J., Amako, K., Apostolakis, J., Arce, P. et al. (2016) 'Recent developments in Geant4', *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 835, pp. 186–225. doi: 10.1016/j.nima.2016.06.125.

3. Rising, M.E., Armstrong, J.C., Bolding, S.R. et al. (2023) 'MCNP code version 6.3.0 release notes', *Tech. Report LA-UR-22-33103*, avalaible at: https://mcnp.lanl.gov/pdf_files/TechReport_2023_LANL_LA-UR-22-33103Rev.1_RisingArmstrongEtAl.pdf (accessed August 8, 2024). doi: 10.2172/1909545.

4. Varignier, G., Fondement, V., Carasco, C. et al. (2023) 'Comparison between GEANT4 and MCNP for well logging applications', *EPJ Web of Conf.*, 288, art. 01002. doi: 10.1051/epjconf/202328801002.

5. Marchais, T., Perot, B., Allinei, P. et al. (2023) 'Experimental validation of a CeBr3 gamma-ray logging probe MCNP model', *EPJ Web of Conf.*, 288, art. 05003. doi: 10.1051/epjconf/202328805003.

6. Cybowska, J. (2023) 'Reactor power measurement using N16 and N17 production rate in MARIA reactor', *EPJ Web of Conf.*, 288, art. 04022. doi: 10.1051/epjconf/202328804022.

7. Nabil, I.M., El-Samrah, M.G., Omar, A., Tawfic, A.F., El Sayed, A.F. (2023) 'Experimental, analytical, and simulation studies of modified concrete mix for radiation shielding in a mixed radiation field', *Sci. Reports*, 13, art. 17637. doi: 10.1038/s41598-023-44978-8.

8. Tseng, W., Furutani, K., Beltran, C., Lu, B. (2024) 'An automation of Monte Carlo workflow for dosimetry study of an Elekta LINAC delivery system in radiotherapy', *TipsRO*, 31, art. 100257. doi: 10.1016/j.tipsro.2024.100257.

9. Ziegler, J., Ziegler, M.D., Biersack, J. (2010) 'The stopping and range of ions in matter', *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 268(11-12), pp. 1818–1823. doi: 10.1016/j.nimb.2010.02.091.

10. Wang, Y., Zeng, X., Decker, J. (2021) 'A GPU-accelerated radiation transfer model using the lattice boltzmann method', *Atmosphere*, 12(10), art. 1316. doi: 10.3390/atmos12101316

11. Askar, T., Yergaliyev, A., Shukirgaliyev, B., Abdikamalov, E. (2024) 'Exploring numba and CuPy for GPU-accelerated Monte-Carlo radiation transport', *Computation*, 12(3), art. 61. doi: 10.3390/computation12030061.

12. Silvestri, S., Pecnik, R. (2019) 'A fast GPU Monte Carlo radiative heat transfer implementation for coupling with direct numerical simulation', *J. of Computational Phys. X*, 3, art. 100032. doi: 10.1016/j.jcpx.2019.100032.

13. Kosov, M.V., Savin, D.I. (2014) *CHIPS-TPT*, Pat. RF, no. 2014611928.

14. Savin, D., Kosov, V. (2017) 'Exclusive data-based modeling of neutron-nuclear reactions below 20 MeV', *EPJ Web of Conf.*, 146, art. 12028. doi: 10.1051/epjconf/201714612028.

15. Thulliez, L., Jouanne, C., Dumonteil, E. (2021) 'Improvement of Geant4 Neutron-HP package: From methodology to evaluated nuclear data library', *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 1027, art. 166187. doi: 10.1016/j.nima.2021.166187.

16. Amadio, G., Apostolakis, J., Bandieramonte, M. et al. (2015) 'The GeantV project: Preparing the future of simulation', *JPCS*, 664, art. 072006. doi: 10.1088/1742-6596/664/7/072006.

17. Amadio, G., Apostolakis, J., Bandieramonte, M. et al. (2015) 'Towards a high performance geometry library for particle-detector simulations', *JPCS*, 608, art. 012023. doi: 10.1088/1742-6596/608/1/012023.

18. Amadio, G., Apostolakis, J., Cosmo, G., Dosaru, D. et al. (2023) 'Offloading electromagnetic shower transport to GPUs', *JPCS*, 2438(1), art. 012055. doi: 10.1088/1742-6596/2438/1/012055.

19. Seibert, S., Latorre, A. (2011) *Fast optical Monte-Carlo simulation with surface-based geometries using Chroma*, avalaible at: https://www.snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF2_CompF0-NF10_NF0_Chroma-045.pdf (accessed August 08, 2024).

20. Kaptanoglu, T., Luo, M., Land, B. et. al. (2020) 'Spectral photon sorting for large-scale Cherenkov and scintillation detectors', *Phys. Review D*, 101, art. 072002. doi: 10.1103/PhysRevD.101.072002.

21. Okada, S., Mirakami, K., Incerti, S. et al. (2019) 'MPEXS-DNA, a new GPU-based Monte Carlo simulator for track structures and radiation chemistry at subcellurar scale', *Med. Phys*., 46(3), pp. 1483–1500. doi: 10.1002/mp.13370.

22. Valassi, A., Yazgan, E., McFayden, J., et al. (2021) 'Challenges in Monte Carlo event generator software for high-luminosity LHC', *Computing and Software for Big Sci*., 5, art. 12. doi: 10.1007/s41781-021-00055-1.

23. Li, Y., Zhang, B., Yang, S., Chen, Y. (2024) 'A data-driven method for calculating neutron flux distribution based on deep learning and the discrete ordinates method', *Energies*, 17(14), art. 3440. doi: 10.3390/en17143440.

24. Gimenez-Alventosa, V., Gomez, V.G., Oliver, S. (2021) 'PenRed: An extensible and parallel Monte-Carlo framework for radiation transport based on PENELOPE', *Computer Physics Communications*, 267, art. 108065. doi: 10.1016/j.cpc.2021.108065.

25. Selau, F.F., Trombini, H., Fadanelli, R.C., Vos, M., Grande, P.L. (2021) 'On the energy-loss straggling of protons in elemental solids: The importance of electron bunching', *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 497, pp. 70–77. doi: 10.1016/j.nimb.2021.03.006.

26. Zyla, P., Barnett, R., Beringer, J. et al. (2020) 'Review of particle physics', *PTEP*, 2020(8), art. 083CO1. doi: 10.1093/ptep/ptaa104.

Авторы

Галюзов Андрей Андреевич [1],
научный сотрудник, AAGalyuzov@vniia.ru
Косов Михаил Владимирович [1],
д.ф.-м.н., главный научный сотрудник,
Kosov@vniia.ru

[1] Всероссийский научно-исследовательский институт автоматики им. Н.Л. Духова,
г. Москва, 127030, Россия

Authors

Andrey A. Galyuzov [1], Research Associate,
AAGalyuzov@vniia.ru
Mikhail V. Kosov [1],
Dr.Sci. (Physics and Mathematics)
Chief Researcher, Kosov@vniia.ru

[1] Dukhov Automatics Research Institute (VNIIA),
Moscow, 127030,
Russian Federation