

АЛГОРИТМ РАСЧЁТА ДОЗЫ НА ОСНОВЕ МЕТОДА МОНТЕ-КАРЛО ДЛЯ ДИСТАНЦИОННОЙ ЛУЧЕВОЙ ТЕРАПИИ

*И.О. Ярышев, А.Н. Моисеев
ООО "РТ7", Москва*

MONTE CARLO METHOD-BASED DOSE CALCULATION ALGORITHM FOR REMOTE RADIATION THERAPY

*I.O. Yaryshev, A.N. Moiseev
ООО "RT7", Moscow, Russia*

Реферат

В лучевой терапии алгоритм расчёта дозы влияет на точность лечения, поэтому необходимо постоянно совершенствовать методику получения дозовых распределений. В данном исследовании продемонстрировано сравнение разработанного упрощённого и адаптированного под фотонную дистанционную лучевую терапию алгоритма на основе метода Монте-Карло с классическим методом Монте-Карло моделирования взаимодействия излучения с веществом, применяемым в Geant4/Gate. В результате анализ гамма-индекса показал хорошее соответствие рассчитанной дозы между алгоритмами при значительном выигрыше разработанного метода в скорости расчёта.

Ключевые слова: лучевая терапия, метод Монте-Карло, расчёт дозы

Abstract

In radiation therapy, the dose calculation algorithm affects the accuracy of treatment, so it is necessary to constantly improve the method of obtaining dose distributions. This study demonstrates a comparison of the developed simplified algorithm based on the Monte Carlo method and adapted for remote radiation therapy with the classical Monte Carlo method for modeling the interaction of radiation with matter used in Geant4/Gate. As a result, the analysis of the gamma index showed a good correspondence of the calculated dose between the algorithms, with a significant gain in the calculation speed of the developed method.

Key words: radiation therapy, Monte Carlo method, dose calculation

E-mail: ilia121608@gmail.com

<https://doi.org/10.52775/1810-200X-2025-107-3-25-29>

Введение

Современная лучевая терапия уже достигла значительных успехов в достижении высокой точности облучения и минимизации воздействия на здоровые ткани. Тем не менее, работа по поиску более совершенных методов расчета дозы продолжается, и одним из возможных путей повышения точности планирования облучения является разработка и модернизация алгоритмов расчета дозы.

В настоящее время существуют различные методики расчета дозы, разработанные разными компаниями, такими как, например, Varian и Elekta, для применения в дистанционной фотонной лучевой терапии совместно со своими аппаратами. Это алгоритмы Acuros XB, AAA, XVMC и др. При этом в данных методах расчета дозы используются различные упрощения для ускорения процесса, так как важна не только точность полученного дозового распределения, но и скорость и ресурсоёмкость алгоритма. Однако вносимые упрощения приводят к различным неточностям при расчете дозы, особенно в гетерогенных средах с разной электронной плотностью и на их границах [1].

Наиболее перспективными с точки зрения точности расчета считаются алгоритмы, основанные на методе Монте-Карло, широко применяемые в физике высоких энергий и доказавшие свою эффективность в этой области. Основным недостатком данного подхода является низкая скорость сходимости, что приводит к очень низкой скорости уменьшения погрешности, порядка $1/\sqrt{N}$, где N – число историй (частиц). В связи с этим вычисления методом Монте-Карло обычно проводят на вычислительных кластерах [2–4]. Однако такой кластер – дорогое устройство, и обеспечить все медицинские центры достаточными вычислительными мощностями не представляется возможным по экономическим причинам.

Выходом из данной ситуации могло бы стать создание оптимизированного алгоритма с разумными упрощениями, основанного на методе Монте-Карло, и последующий перенос расчета на графические ускорители. Реализации такого алгоритма и было посвящено данное исследование.

Материал и методы

Реализация алгоритма осуществлялась на языке программирования C# [5], как на од-

ном из наиболее распространенных и универсальных языков. Для упрощения и ускорения расчета в алгоритме не учитывалось распространение тормозного излучения вторичных частиц. Таким образом, энергия радиационных потерь поглощалась локально в воксел, соответствующий положению частицы в пространстве. Верификация метода проводилась путём сравнения результатов вычислений разработанного алгоритма с результатом расчета аналогичной геометрии в среде моделирования переноса излучения Gate [6]. Визуализация и анализ результатов проводилось при использовании библиотек matplotlib [7], pandas [8], numpy [9] языка Python [10].

Первым этапом верификация алгоритма проводилась на фантоме, выполненном в виде параллелепипеда воды со сторонами 7,5 см по оси Y, 14,5 см по оси Z и 15,25 см по оси X, с воздушным отступом по 0,5 см от торцов вдоль оси Z. Излучение в виде моноэнергетического (1,25 МэВ) пучка фотонов с поперечным сечением в виде окружности диаметром 3 см направлено перпендикулярно фантому вдоль оси Z фантома. На глубине 3 см от поверхности воды располагался слой в виде материала, аналогичного воде, но с электронной плотностью равной $0,116 \text{ г/см}^3$ (выступает в роли имитации лёгких). Размер вокселя составлял 0,05 см. Число фотонов для расчета дозы выбиралось равным 10^7 . В результате работы алгоритма был получен профиль дозы вдоль оси Z фантома при суммировании в плоскости XY вокселей матрицы 5×5 . Данная зависимость дозы от глубины сравнивалась с аналогичной, рассчитанной в Gate. Схема фантома в виде 3D изображения из Gate представлена на рис. 1.

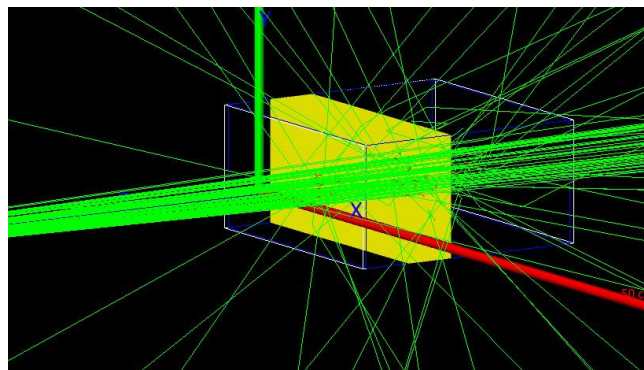


Рис. 1. Схема фантома для первичной верификации с падающим пучком. Жёлтым цветом обозначен слой воды с изменённой электронной плотностью

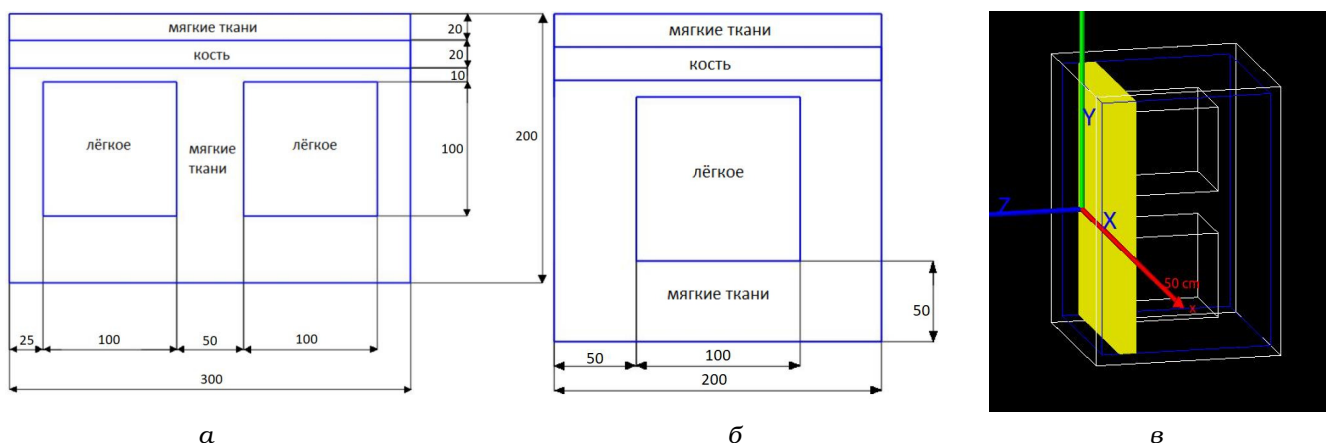


Рис. 2. Схема фантома для верификации алгоритма на втором этапе: а – вид спереди; б – вид сбоку; в – 3D изображение (размеры на схемах а и б представлены в мм)

На втором этапе верификации были получены трёхмерные дозовые распределения на антропоморфном фантоме в виде грудной клетки с лёгкими и костной структурой перед ними, материалы которого соответствовали данным из базы NIST [11]. Схема с характерными размерами в мм, а также 3D картинка, полученные в Gate, представлены на рис. 2. Моноэнергетический (1,25 МэВ) пучок фотонов с диаметром 100 мм направлялся либо перпендикулярно фантому вдоль его оси с изоцентром на поверхности фантома, либо под углом гантри 150°, при этом изоцентр находился в центре фантома. Размер вокселя – 1 мм по оси фантома и его меньшей главной оси, 1,5 мм по второй наибольшей главной оси. Количество фотонов составляло 10 миллионов.

Для верификации дозовых распределений на втором этапе использовался метод локального гамма-индекса [12]. Расчет проводился с помощью библиотеки `pymedphys` [13] со следующими параметрами: порог по дозе – 2 %; порог по расстоянию – 1 мм; отсечка дозы – 20 %; параметр интерполяции – 10; максимальное значение гамма-индекса – 2. Чтобы избежать влияния шума, способного искусственно занижать показания гамма-индекса, и получить наиболее объективную оценку точности разработанного алгоритма, применялось сглаживание посредством фильтра Гаусса библиотеки `scipy` [14] со следующими параметрами: среднее квадратическое отклонение – 100, радиус сглаживания – 3.

Результаты и обсуждение

На рис. 3 представлены графики зависимости дозы на оси фантома, полученные для разработанного алгоритма и Gate-a, из которых можно увидеть, что предложенная методика расчёта дозы на основе метода Монте-Карло демонстрирует практически полное совпадение с классическим алгоритмом, представленным в Gate.

Отклонение результатов расчёта дозы от референсных значений на границах раздела сред обусловлено особенностью трассировки частиц в воксельной геометрии, а также возможными различиями между оригинальным алгоритмом и Gate-ом в задании характеристик материалов. Тем не менее, дальнейшее сравнение с применением гамма-анализа показало, что данное отклонение не превышает допустимых значений.

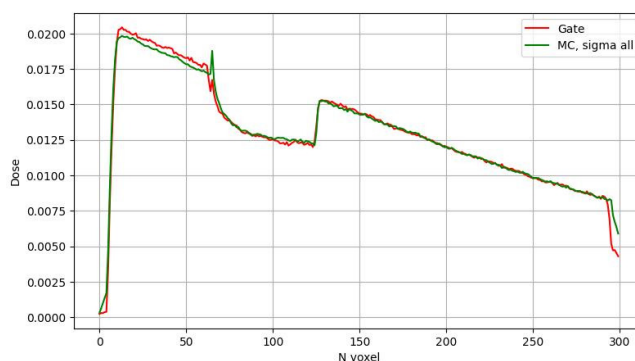


Рис. 3. Сравнение профилей дозового распределения вдоль оси фантома, полученных посредством разработанного алгоритма и инструментария Gate

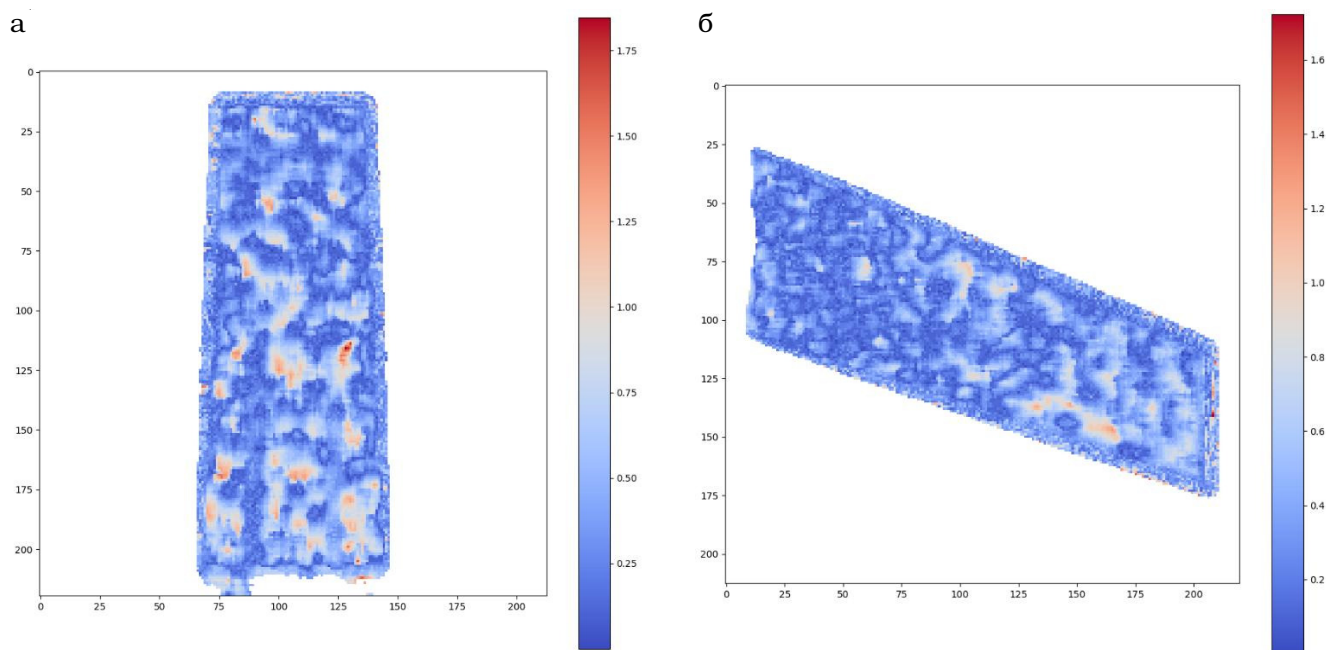


Рис. 4. Результаты расчёта гамма-индекса γ (2 %, 1 мм) при сравнении трёхмерных дозовых распределений, полученных при помощи разработанного алгоритма и Gate (срез на оси фантома): а – перпендикулярное падения пучка на фантом (поворот гантри 0°); б – поворот гантри на 150°

Так, рис. 4 показывает срез вдоль оси рассчитанных значений гамма-индекса исследуемых дозовых распределений. Из полученных метрических оценок ясно, что сравниваемые дозовые распределения совпадают с точностью до отдельных вокселей, расхождения в которых объясняются случайной погрешностью.

Стоит также отметить, что расчёт дозы для 10 млн фотонов без применения методов параллельных вычислений для разработанного алгоритма занимал около 10–15 минут, тогда как аналогичное моделирование взаимодействия с фантомом такого же числа частиц в Gate занимало примерно четыре часа. Введение технологии переноса вычислений на видеокарту посредством использования библиотеки ILGPU [15] сократило время, необходимое на расчёт 10 млн историй в условиях выбранной геометрии, до 25 секунд. Алгоритм позволяет рассчитывать 100 млн фотонов менее чем за 1,5 минуты. Таким образом, разработанный механизм расчета при сохранении точности метода Монте-Карло моделирования взаимодействия фотонного излучения с веществом гетерогенной среды обеспечивает практически 600-кратное ускорение при его исполнении на графических процессорах по сравнению с алгоритмом, используемым в Gate.

Заключение

В данном исследовании был представлен разработанный алгоритм на основе метода Монте-Карло, специализированный и оптимизированный для задач дистанционной фотонной лучевой терапии. Сравнение дозовых распределений, полученных при помощи новой методики расчёта и классического алгоритма, реализованного в Gate-e, показало высокую степень качественного соответствия результатов работы двух алгоритмов. Некоторые количественные различия предположительно обусловлены особенностями воксельной геометрии, случайными погрешностями, а также неучтенным вкладом тормозного излучения.

Значимым фактом также является то, что новый алгоритм при использовании распараллеливания и технологии CUDA оказался практически в 600 раз быстрее расчёта в Gate.

Список литературы

1. Woon YL, et al. Comparison of selected dose calculation algorithms in radiotherapy treatment planning for tissues with inhomogeneities. *Journal of Physics: Conference Series*, 694, 012024 (2016). <https://doi.org/10.1088/1742-6596/694/1/012024>.

2. Andreo P. Monte Carlo techniques in medical radiation physics. *Physics in Medicine and Biology* 1991; 36: 861-920. <https://doi.org/10.1088/0031-9155/36/7/001>.
3. Bielajew A. History of Monte Carlo. In: Seco J, Verhaegen F, eds. *Monte Carlo Techniques in Radiation Therapy*. Boca Raton: CRC Press; 2013. p. 3-16. <https://doi.org/10.1201/b13961>.
4. Andreo P. Monte Carlo simulations in radiotherapy dosimetry. *Radiation Oncology* 2018; 13 (1): 121. <https://doi.org/10.1186/s13014-018-1065-3>.
5. Microsoft Corporation. C# Language Documentation. Online resource available at: <https://learn.microsoft.com/ru-ru/dotnet/csharp>.
6. Jan S, Benoit D, Becheva E, et al. GATE V6: a major enhancement of the GATE simulation platform enabling modeling of computed tomography and radiotherapy. *Physics in Medicine and Biology* 2011; 56 (4): 881-901. <https://doi.org/10.1088/0031-9155/56/4/001>.
7. Hunter JD. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 2007; 9 (3): 90-5. <https://doi.org/10.1109/MCSE.2007.55>.
8. McKinney W. Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference* 2010; Vol. 445. <https://doi.org/10.25080/Majora-92bf1922-00a>.
9. Harris CR, Millman KJ, van der Walt SJ, et al. Array Programming with NumPy. *Nature* 2020; 585: 357-62. <https://doi.org/10.1038/s41586-020-2649-2>.
10. Python Official Website. Online resource available at: <https://www.python.org/>.
11. Coursey JS, Schwab DJ, Tsai JJ, Dragoset RA. Atomic weights and isotopic compositions (Version 4.1). Online resource available at: <http://physics.nist.gov/comp>.
12. Das S, Kharade V, Pandey VP, et al. Gamma index analysis as a patient-specific quality assurance tool for high-precision radiotherapy: a clinical perspective of single institute experience. *Cureus* 2022; 14 (10): e30885. <https://doi.org/10.7759/cureus.30885>.
13. Biggs S, Jennings M, Swerdloff S, et al. PyMedPhys: A Community Effort to Develop an Open, Python-Based Standard Library for Medical Physics Applications. *Journal of Open Source Software* 2022; 7 (78): 4555. <https://doi.org/10.21105/joss.04555>.
14. Virtanen P, Gommers R, Oliphant TE, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 2020; 17: 261-72. <https://doi.org/10.1038/s41592-019-0686-2>.
15. Koester M. A modern GPU compiler for .NET programs. Online resource available at: <https://ilgpu.net>.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Финансирование. Исследование не имело спонсорской поддержки.

Участие авторов. Статья подготовлена с равным участием авторов.

Поступила: 07.07.2025. Принята к публикации: 10.09.2025.

Conflict of interest. The authors declare no conflict of interest.

Financing. The study had no sponsorship.

Contribution. Article was prepared with equal participation of the authors.

Article received: 07.07.2025. Accepted for publication: 10.09.2025.