Published in final edited form as:

Phys Med Biol. ; 67(17): . doi:10.1088/1361-6560/ac8716.

MOQUI: An open-source GPU-based Monte Carlo code for proton dose calculation with efficient data structure

Hoyeon Lee[1], **Jungwook Shin**[2], **Joost M. Verburg**[1], **Mislav Bobić**[1,3], **Brian Winey**[1], **Jan Schuemann**[1], **Harald Paganetti**[1]

[1]Dept. of Radiation Oncology, Massachusetts General Hospital and Harvard Medical School, Boston, MA 02114, USA

[2]Division of Cancer Epidemiology and Genetics, National Cancer Institute, National Institutes of Health, Rockville, MD 20850, USA

[3]Department of Physics, ETH, Zürich 8092, Switzerland

Abstract

Objective: Monte Carlo (MC) codes are increasingly used for accurate radiotherapy dose calculation. In proton therapy, the accuracy of the dose calculation algorithm is expected to have a more significant impact than in photon therapy due to the depth-dose characteristics of proton beams. However, MC simulations come at a considerable computational cost to achieve statistically sufficient accuracy. There have been efforts to improve computational efficiency while maintaining sufficient accuracy. Among those, parallelizing particle transportation using graphic processing units (GPU) achieved significant improvements. Contrary to the central processing unit (CPU), a GPU has limited memory capacity and is not expandable. It is therefore challenging to score quantities with large dimensions requiring extensive memory. The objective of this study is to develop an open-source GPU-based MC package capable of scoring those quantities.

Approach: We employed a hash-table, one of the key-value pair data structures, to efficiently utilize the limited memory of the GPU and score the quantities requiring a large amount of memory. With the hash table, only voxels interacting with particles will occupy memory, and we can search the data efficiently to determine their address. The hash-table was integrated with a novel GPU-based MC code, moqui.

Main results: The developed code was validated against an MC code widely used in proton therapy, TOPAS, with homogeneous and heterogeneous phantoms. We also compared the dose calculation results of clinical treatment plans. The developed code agreed with TOPAS within 2%, except for the fall-off and regions, and the gamma pass rates of the results were >99% for all cases with a 2mm/2% criteria.

Significance: We can score dose-influence matrix and dose-rate on a GPU for a 3-field H&N case with 10 GB of memory using moqui, which would require more than 100 GB of memory with the conventionally used array data structure.

hlee80@mgh.harvard.edu .

Keywords

Monte Carlo; Proton therapy; Graphic-processing-unit

1. INTRODUCTION

Due to proton beams' depth dose characteristics, proton therapy can be delivered with a highly conformal tumor dose and with a lower integral dose compared to photon therapies. In order to fully utilize this advantage of proton therapy, it is essential to accurately calculate the proton range. Two dose calculation algorithms are commonly used in the clinic, pencil beam algorithms and Monte Carlo (MC) algorithms. Though typical pencil-beam algorithms may increase range uncertainties by about 2% compared to Monte Carlo (MC) dose calculation algorithms (Paganetti 2012), the long computational time of MC algorithms discourages their application in routine clinical practice, particularly for inverse planning.

Several studies have been conducted to improve the computation efficiency of MC algorithms while maintaining dose calculation accuracy. The solutions involved simplified physics models that only consider the dominant physics processes occurring in the human body (Fippel and Soukup 2004, Kohno *et al* 2002), sample primary particles based on the importance of beams (Li *et al* 2015), reformulate restricted stopping power calculation (Maneval *et al* 2017), or parallelized particle transportation using multi-threading techniques (Souris *et al* 2016). Further, there have been efforts to reduce computation time using parallelization algorithms and graphic processing units (GPUs) (Jia *et al* 2012, Yepes *et al* 2010, Qin *et al* 2016, Kohno *et al* 2011, Maneval *et al* 2019). GPU-based MC codes can significantly improve efficiency with clinically acceptable dose calculation accuracy (Gajewski *et al* 2021, Giantsoudi *et al* 2015, Tseung *et al* 2015, Fracchiolla *et al* 2021). This enabled fast MC-based proton treatment optimization (Ma *et al* 2014, Li *et al* 2016), fast 4D dose calculation (Pepin *et al* 2018), and online plan adaptation to mitigate the uncertainty of anatomical changes in patients (Bobić *et al* 2021). Recently, GPU-based MC has been deployed with clinical treatment planning systems to improve the efficiency in the clinical workflow (Fracchiolla *et al* 2021, Lin *et al* 2017).

One downside of using GPU is the memory limitation. A GPU has a smaller memory size than the random-access memory (RAM) available in a CPU. This is a major hurdle when scoring spatial or temporal quantities, e.g., dose-influence matrix (D_{ij} matrix) or dose rate. For example, when scoring a D_{ij} matrix with the array, the required memory is equal to the number of voxels in the scoring grid times the number of beamlets. To run such an application on a GPU, one can either limit the number of voxels in the scoring grid or split beamlets in a beam field into multiple runs. The former approach cannot provide a full D_{ij} matrix for optimization. The latter approach will increase the runtime due to the frequent data transfer between CPU memory and GPU memory.

To address this issue, we developed a novel memory-efficient GPU-based MC code, MOnte Carlo code for QUIck proton dose calculation (moqui). Moqui employs a key-value pair data structure for the scoring system to improve memory efficiency.

2. Methods and Materials

2.A. Physics model and material data

We employed a simplified physics model developed by Fippel and Soukup (Fippel and Soukup 2004). The model considers electromagnetic, ionization, proton-oxygen inelastic, proton-oxygen elastic, and proton-proton elastic processes. To determine the interaction of particles, we consider the maximum step size of a particle (l_{max}), distance to the closest boundary of voxel (l_b), and the mean free path (MFP) of each interaction (l_{MFP}).

l_{max} was set to 1 mm in this study. The closest distance to the boundary is calculated using the axis-aligned bounding boxes and a ray boundary intersection algorithm (Williams *et al* 2005). The algorithm calculates the travel length to collide with each plane in a bounding box from the current position and direction of a particle using Eq. (1), where d is the distance, b is the position of a plane, p is the current position of the particle, e is the direction of the particle, $\overrightarrow{n_i}$ is a unit vector parallel to the i-th axis, and · is the inner product between vectors. The shortest positive distance among 6 traveling distances to be selected for l_b is then

$$d = \frac{(b - p) \cdot \overrightarrow{n_l}}{e \cdot \overrightarrow{n_l}} \tag{1}$$

The l_{MFP} is sampled using fictitious interaction methods (Jia *et al* 2012, Salvat *et al* 2009) with cross-section data of ionization and nuclear interactions, which were taken from Geant4 v10.6.p03 (Agostinelli *et al* 2003) and shown in Figure 1. We took the ionization cross-section data from G4BraggModel for low energy protons (< 2MeV) and G4BethBlochModel for high energy protons (> 2MeV). Cross-section data of the elastic and inelastic interactions were taken using the QGSP_BERT_HP physics list. The particle moves to the shortest distance among l_{max}, l_b, and l_{MFP}.

To calculate the energy loss in a step, we first convert the length of a step in a voxel (Δl) into length in water (Δl_w) using the relative stopping power of the material to water (RSP). The energy loss of the step is sampled from a Gaussian distribution to account for energy straggling. The average of the distribution is obtained by subtracting the current particle energy from the energy at the end of the step. The energy of the particle after the step $\left(T_p^{end}\right)$ is calculated using Eq. (2), where T_p is the current energy of a particle, T_e^{min} is the minimum energy of electrons (0.0815 MeV in this study), and $L_w\left(T_p, T_e^{min}\right)$ is the restricted stopping power in water at energy T_p with T_e^{min}. We calculate the residual range after the step $\left(\int_0^{T_p} \frac{dT'_p}{L_w\left(T'_p, T_e^{min}\right)} - \Delta l_w\right)$ and obtain the proton energy with the residual range using linear interpolation. The standard deviation of the distribution is calculated as described in Fippel and Soukup (Fippel and Soukup 2004).

$$\Delta l_w = \int_{T_p^{end}}^{T_p} \frac{dT'_p}{L_w\left(T'_p, T_e^{min}\right)} = \int_0^{T_p} \frac{dT'_p}{L_w\left(T'_p, T_e^{min}\right)} - \int_0^{T_p^{end}} \frac{dT'_p}{L_w\left(T'_p, T_e^{min}\right)} \quad (2)$$

If l_{MFP} was the shortest distance, the energy loss of the proton is followed by either ionization, proton-proton elastic interaction, proton-oxygen elastic interaction, or proton-oxygen inelastic interaction. The interaction is randomly selected based on the cross-section data.

We follow the model described by Fippel and Soukup to sample the secondary particle energy for each process. Among the secondary particles, protons are transported the same way as the primary protons. The energy of short-range particles, e.g., electrons, is locally deposited, and long-range particles, such as neutrons and photons, are ignored. In low-density materials, like air, the secondary charged particles would travel to other voxels until their energy drops below the minimum energy rather than deposit their energy locally. However, since we don't transport them, it could result in unreasonably high dose in low-density voxels. Therefore, we exclude the energy deposition from secondary charged particles in low-density materials. This would not impact the dose calculation results of the cases in this study because the low-density region is not a region of interest for the cases. This physics model approach has shown good agreement with dose calculation results that consider comprehensive physics processes for proton dose calculation (Jia *et al* 2012, Fippel and Soukup 2004, Giantsoudi *et al* 2015).

The cross-section and electron density are obtained for water and scaled by mass density for different materials. We then calculate material characteristics based on the mass density of each material. The mass density of each voxel is calculated from the Hounsfield unit (HU) using the conversion method proposed by Schneider, et al. (Schneider *et al* 2000). The radiation length as a function of mass density and RSP as a function of mass density and energy for each material are defined as functions obtained by curve-fitting of data taken from Geant4 (Agostinelli *et al* 2003). The density conversion curve, radiation length, and RSP curve per density of materials, and the curve fitting results are presented in Figure 2. Water with 75 eV ionization energy was used to obtain the RSP curve.

Moqui transports particles in multiple volumes sequentially if there is more than one volume in the geometry. This is to consider objects between the source and the patients, such as range shifters. Particle transport is performed with the same physics models in all volumes.

2.B. Code structure

All components in moqui, such as particle source, geometric component, and scoring system, are defined as C++ classes that follow a modular design approach capable of adding new features in the future. The code structure of moqui is summarized in Figure 3. The environment and treatment machine are user-defined classes. The "environment" class includes scorer selection, functions to read the patients' CT image and RT-Ion plan. A conversion function from HU to material mass density and a facility-specific beam model, which includes spot size, angular spread of spots, energy spread, and the number of primary particles per monitor unit (MU), also referred as absolute dosimetry, are included in the “treatment machine” class. The parameters in the beam model are optimized to match MC calculation results and measured data for the proton machine at our institution. These parameters are necessary to relate DICOM-RT-Ion information to treatment delivery settings. We use a previously developed DICOM-RT-Ion interface (Shin *et al* 2020), which reads machine names, primary particles’ data, and a range shifter definition from RT-Ion PLAN data.

The RT-Ion PLAN contains the nominal energy and the 2D mean position of each beamlet on beam’s eye view. The number of primary particles is calculated from a facility-specific conversion of MU to the number of particles. The position and direction of each primary particle are sampled based on the defined spot size, energy spread, and angular spread. The sampled primary particles are stored in a STACK from which they are transported through the geometry following the physics process until their energy drops below the minimum energy (T_p^{min}, we used 0.5 MeV for this study). The transportation function ends when no particle is left in the STACK. After transportation, scoring results are copied back to the CPU for post-processing and are subsequently saved.

The GPU implementation was done using compute unified device architecture (CUDA v10.2) (Luebke 2008), which enables parallel computing on GPUs from NVIDIA (Santa Clara, CA, USA) using the C/C++ programming language. There are different types of memory on a GPU, i.e., global memory, shared memory, constant, and texture memories. The global, texture, and constant memories are accessible by all threads, while the shared memory is accessible by a block of threads. In our implementation, we used global and constant memory only. The cross-section data are defined in constant memory since these are read-only data with small size. The other data required in particle transport, such as patient images and particle data, are stored in global memory since they need to be accessible from all threads. The hash-table is also defined in the global memory to make it accessible by all threads.

2.C. Data structure for the scoring system

For the scoring system, we adopted the hash table (Maurer and Lewis 1975), one of the key-value pair data structures. Each element in the hash table comprises two keys and a value, as shown in Figure 4.

In a conventional data structure, i.e., the array structure, every scoring voxel needs to be assigned to each array element for scoring. For example, suppose we try to score dose deposition of 3 different beamlets to 5 voxels. In that case, array elements need to be assigned for each voxel, including voxels that have not interacted with particles, as shown in Figure 5 (a). However, one-to-one mapping between the scoring voxel and the element in the data structure is not necessary for the hash-table data structure since a hash function determines the index of an element from keys. Therefore, voxels that do not interact with the particles will not occupy memory in the key-value data structure, and we can efficiently score quantities that require a large memory size, as shown in Figure 5 (b). Though there are only 4 interacting voxels from 3 beamlets in Figure 5 (a), we need to allocate 15 elements for scoring with the array, while the hash-table only needs 4 elements in Figure 5 (b). The hash-table elements with NULL will be used if there are more interactions. Therefore, the hash-table improves the memory utilization efficiency and can store more elements with the same memory constraints compared to the array. The hash-table stores data in an unsorted manner; however, the time required to search over the data is independent of data size since the hash function returns indices from given keys (Maurer and Lewis 1975).

In our hash-table implementation, one of the keys represents the voxel index of the scoring grid, and the other represents a user-defined index, such as the spot index of a beam or temporal index of a beam. We use a Murmur hash function (Farrell 2020, Appleby 2011). The Murmur hash function takes a key and a seed. The function multiplies a constant value, bit-wise shift, and bit-wise exclusive or (XOR) operation to generate a hash value, an index of memory, from the given integers. The voxel index is used as the key and the user-defined index is used as the seed for the hash function.

The CPU implementation of the hash-table can usually be expandable on demand. However, expanding the size of the data structure is inefficient and time-consuming on GPU. Therefore, we pre-allocate memory for the hash table at the beginning of the simulation to avoid such inefficiency. To select an address of the memory without exceeding the predefined allocation, modulo of the hash value and the maximum size of the address are calculated. Due to the modulo operation, we may get the same address for different input keys. If a different element already took the address, we use the linear probing method (Pagh *et al* 2009) to find an empty address. The linear probing method searches over the hash table from the initial address by moving 1 address at a time until a vacancy is found.

2.D. Available quantities for scoring

In our default implementation, moqui supports scoring energy deposition, dose-to-water (D_w), and dose-to-medium (D_m). It can also score various other quantities, including dose-weighted linear energy transfer (LET_d), track-weighted linear energy transfer (LET_t), dose-influence matrix (D_{ij} matrix) (Unkelbach *et al* 2013), and dose rate. The D_w is a quantity that is used for decision making in clinical practice of radiotherapy (Andreo 2015, Paganetti 2009), D_m is a quantity that could be used for clinical decision making in proton therapy (Paganetti 2009) while LET_d or LET_t are quantities used to analyze or create plans considering biological effects (Giantsoudi *et al* 2013, Unkelbach *et al* 2016). These quantities are commonly used in proton therapy and most proton MC packages support them (Perl *et al* 2012, Schiavi *et al* 2017, Tseung *et al* 2015, Jia *et al* 2012). The D_{ij} matrix and dose-rate can be used for plan optimization and are implemented to demonstrate the improved memory efficiency when using the hash-table.

We use Eq. (3) to score D_w from the energy deposition to a voxel v. The fitted curves of RSP are used to convert the step length in a voxel to a step length in water described in Section 2. A.

$$D_w(v) = \sum_j \frac{dE_j \times 1.60218 \times 10^{-13}}{\rho(v)V(v)RSP\left(\rho(v), T_p\right)} \tag{3}$$

We use Eq. (4) and (5) to score LET_d (Grassberger *et al* 2011). In the equations, dE_{ij} is the energy deposited in voxel i from spot j, l_{ij} is the length of the step in a voxel i from a spot j, ρ_i is the mass density of the i-th voxel, $LET_{d_i}^F$ is the dose-weighted LET in voxel i for beam field F, and D_i^F is the dose delivered to voxel i for beam field F. For LET_d scoring, we set an upper threshold of 25 MeV/mm for dE_{ij}/l_{ij} to exclude spurious spikes caused by an artificial step limitation of the protons in the LET_d distribution (Granville and Sawakuchi 2015, Cortés-Giraldo and Carabe 2015).

$$LET_{d_i}^F\left(Mev/mm/\left(g/cm^3\right)\right) = \frac{\sum_j \left(dE_{ij} \times \left(dE_{ij}/l_{ij}\right)\right)}{\rho_i \sum_j dE_{ij}} \qquad (4)$$

$$LET_{d_i}\left(Mev/mm/\left(g/cm^3\right)\right) = \frac{\sum_F LETd_i^F \times D_i^F}{\sum_F D_i^F} \qquad (5)$$

Dose-averaged dose-rate (DADR) (Water *et al* 2019) was implemented to demonstrate dose-rate scoring in moqui based on Eq. (6). In the equation, d_{ij} is dose deposited in voxel i from spot j, t_j is the beam-on time of spot j, and D_i is the dose deposited in voxel i. In this study, we assumed that protons are generated at a constant rate (12.483× 10^{12} protons/second), i.e., the same number of primary particles are generated regardless of energy.

$$\dot{D}_l(Gy/s) = \sum_j \frac{d_{ij} \times \left(d_{ij}/t_j\right)}{D_i} \qquad (6)$$

2.E. Validation and test cases

We compared measured integrated depth-dose (IDD) curves of D_w to validate the physics model used in moqui and compared patient dose calculation results. For comparison, we ran the same simulations in TOPAS version 3.6.p1 (Perl *et al* 2012), a Geant4-based MC package widely used for research purposes. In TOPAS, we used 'HadronPhysicsQGSP_BERT_HP', 'G4HadronElasticPhysicsHP', 'G4IonBinaryCascadePhysics', 'G4DecayPhysics', 'G4StoppingPhysics', and 'G4EMStandardPhysics_option4' physics modules. The production cuts in the simulations were selected empirically to balance the number of secondary particles generated during the simulation with the accuracy of the dose calculation. The production cuts smaller than half of the voxel size changed results no larger than the statistical uncertainties with a voxel size of 1mm^3. Therefore, we used 0.4 mm production cuts for particles except for electrons. The electron production cut was set to 0.1 mm.

We first simulated mono-energetic proton sources impinging on water and inhomogeneous phantoms. The inhomogeneous phantom represents blocks of water (HU: 0), lung (HU: −700), and bone (HU: 500) (Figure 6). The homogeneous water phantom has the same dimension filled with water (HU: 0). Mono-energetic protons were uniformly sampled from a 6*6 cm^2 sized square source centered around the center of the phantom. The source was positioned 0.5 cm in front of the phantom with proton energies of 50 MeV, 100 MeV, 150 MeV, and 200 MeV. We simulated 50 million particles per batch and added batches until the simulation achieved the selected statistical uncertainty.

Subsequently, we compared the dose calculation results for patient cases using CT images. We created proton therapy plans for prostate, liver, and H&N cases using RayStation (RaySearch Laboratories, Stockholm, Sweden). Treatment data of patient cases used in this study are summarized in Table I. The H&N plan has a range shifter made of lucite with density 1.191 g/cm^3. The plans and CT images were exported in DICOM format and were provided to TOPAS and moqui to calculate the D_w distribution to the patients using the same beam model in both MC systems. In this comparison, we set the scoring grid to have the same size as the CT grid.

We used the batch-based method (Rogers *et al* 2021) to calculate statistical uncertainty. The method runs multiple batches of particles and calculated uncertainty using Eq. (7), where v are the voxels, N is the number of batches, $X_i(v)$ is the scored quantity at voxel v in batch i, $\overline{\mathrm{X}}(v)$ is an average value of quantities at voxel v. We averaged the relative uncertainties $(s/\overline{X})$ of voxels receiving more than 10% of the maximum dose for phantom cases, and more than 50% of the maximum dose per field for patient cases. In all cases, the statistical uncertainty was set to 1% for both TOPAS and moqui. Though there are drawbacks to the batch-based methods (Walters *et al* 2002), the method was used because of its simplicity.

$$s(v) = \sqrt{\frac{\sum_{i=1}^{N}\left(X_i(v) - \overline{X}(v)\right)^2}{N(N-1)}} \tag{7}$$

We used the “DoseToMaterial” scorer and water with an ionization potential of 75 eV as material to obtain dose-to-water in TOPAS and turned on the “PreCalculateStoppingPowerRatios” option to improve the computational efficiency. To score LET_d in TOPAS, we used the “ProtonLET” scorer. In all validation cases, no variance reduction technique was used, and no post-processing was applied for dose-to-water scoring. For LET scoring, numerator and denominator in Eq. (4) were scored separately, and they were combined in the post-processing step.

Moqui was run on a workstation equipped with Intel Core i9–9820X, 64GB of RAM, and NVIDIA GeForce RTX 2080 with 10 GB of memory. TOPAS was run on a CPU research cluster with a multi-threading of 3 CPUs.

3. RESULTS

3.A. Validation

The IDD curves of proton sources with energies of 100 MeV and 200 MeV in water phantom are shown in the top left and the bottom left of Figure 7, respectively. For the inhomogeneous phantom, we integrated dose passing through the bone and lung regions separately, and the results are shown in the middle and the right in Figure 7, respectively.

We also plotted lateral profiles of dose deposition with 100 MeV primary protons at two different depths in the inhomogeneous phantom (76 mm and 81 mm from the entrance) in Figure 8. The lung insert and the bone insert are placed in the negative and positive parts of the phantom in the figure, respectively. The relative differences between the IDDs of TOPAS and moqui are within 2%, except for the fall-off region, and the range difference was smaller than 0.3 mm for every energy, which is smaller than a voxel size. The discrepancy in the distal fall-off region was caused by sub-voxel level differences in the range and the steep gradient. The difference in the range between two MC packages is similar to what has been reported in the literature when using the physics model used in moqui (Fippel and Soukup 2004). We performed local gamma evaluation for the 3D dose distribution with 2mm/2% and 1mm/1% criteria. We also performed evaluations with 0mm/2% criteria since both results are simulations and there is no setup error. The gamma pass rates are summarized in Table II.

3.B. Dose calculation results based on patient treatment plans

Representative slices of the 3D dose distribution for the patient simulations from the two MC codes and their relative differences are shown in Figure 9 with the corresponding CT slices. The relative difference was calculated using Eq. (8), where $D_{TOPAS}(v)$, $D_{moqui}(v)$, and D_{TOPAS}^{max} are the TOPAS dose at voxel v, moqui dose at voxel v, and maximum value in dose distribution from TOPAS. We also compared the dose-volume histogram (DVH) for each case (shown in Figure 10). The DVHs from moqui and TOPAS agree well within dose calculation uncertainties. For quantitative comparison, we performed gamma evaluation for the 3D dose distributions with 2mm/2%, 0mm/2%, and 1mm/1% criteria, and the results are summarized in Table III. As summarized in the figures and table, moqui shows good agreement with TOPAS.

$$\Delta D(v) = \frac{D_{TOPAS}(v) - D_{moqui}(v)}{D_{TOPAAS}^{max}} \times 100(\%) \tag{8}$$

The average runtimes to calculate a batch of histories per beam of each patient case are summarized in Table IV. Moqui took less than one minute to calculate the dose for one beam, while it took more than 5 hours with TOPAS with the hardware environment mentioned earlier. However, the runtime comparison between moqui and TOPAS cannot be done on the same computational environment. Besides, both MC packages handle secondary particles in a different way and use different physics models. To eliminate effects of the secondary particle handling, we ran TOPAS without secondary particle transportation, and the results are summarized in Table IV with other results. Ignoring secondary particle transportation in TOPAS reduced the runtime by 32%~61% (depending on the case) without significant dose calculation accuracy reduction.

3.C. Dose-averaged LET, Dose-influence matrix, and Dose-rate scoring

CT slices with results for LET_d for the H&N case from TOPAS and moqui are shown in Figure 11 (a). The LET_d-volume histogram is also shown in Figure 11 (b) for comparison. The LET_d calculation results from moqui show good agreement with TOPAS. We needed to obtain both LET_d and dose distribution simultaneously to calculate the distribution using Eq. (5) and (6). Scoring two quantities at the same time does not have a significant impact on the runtime of TOPAS, while it increased the runtime of moqui from 8 to 23 seconds.

A sparse map of the D_{ij} matrix of the H&N case is shown in Figure 12 (a). We sampled 0.1% of the values over the 70th percentile for visualization purposes. The plan consists of 3 fields, and each field has about 2700 spots. We scored the D_{ij} matrix for all voxels inside the patient's skin, and there are 7,502,788 voxels inside the patient volume. To score the D_{ij} matrix using the array data structure, we would require 147.3 GB ~162.8 GB of memory per field, depending on the number of spots. By contrast, in our implementation, we scored the D_{ij} matrix with a 10 GB memory GPU. We used about 8k primary protons per spot for the scoring, and it took about 3 minutes to the transport particles. The DADR can also be scored in moqui using Eq. (6), as shown in Figure 12 (b).

4. Discussion

Moqui showed good agreement with TOPAS and has similar accuracy compared to the existing GPU-based MC codes using the same physics model in terms of 2mm/2% gamma evaluation results (>99% for phantom cases) (Jia *et al* 2012, Schiavi *et al* 2017). In terms of runtime, moqui can calculate dose and LET_d distributions in patients more efficiently than TOPAS while it is less efficient than previously developed GPU-based MC packages with the same physics model (Schiavi *et al* 2017). Though the comparison was not done on the same environment, moqui took 18 seconds to track 10 million 150 MeV primary protons in a water phantom on NVIDIA GTX 2080 Ti, while the MC package by Schiavi *et al* took 1 second on NVIDIA GTX 1080. We are currently analyzing our code and memory management load to improve its efficiency. However, improved memory efficiency of moqui enables scoring quantities which require large memory on a single GPU. As pointed out previously by others (Li *et al* 2016, Ma *et al* 2014), memory limitation is a major bottleneck when scoring a D_{ij} matrix on a GPU and might require multi-GPU implementation for cases with large target volumes or multiple target volumes to minimize the latency due to the data transfer between CPU and GPU. Moqui scores D_{ij} matrices for such cases with fewer GPUs or even on a single GPU without substantial computational performance loss. Thus, the advantages of moqui compared to other GPU MC packages are (1) the scoring system with the hash-table data structure to improve the memory efficiency of the GPU enabling scoring the D_{ij} matrix or dose-rate on a single GPU and (2) the open-source distribution of the code allowing others to add new physics models or scoring quantities.

For further improvements in computational efficiency, equivalent restricted stopping power (L_{eq}) (Maneval *et al* 2017) could be implemented in moqui. The method reformulates the equation to calculate energy loss of protons regardless of the maximum step length and eliminates the effect of the number of voxels in the energy loss sampling. The method showed promising results in reducing calculation time by about 50% with similar dose calculation accuracy. Additionally, utilizing texture memory on GPU to store read-only data and utilize an interpolation function provided by CUDA could help improving the efficiency as it is utilized in the other GPU-based MC packages. (Jia *et al* 2012, Schiavi *et al* 2017, Maneval *et al* 2019)

The current implementation has some limitations. In our physics model, we consider that the energy of all charged secondary particles except for protons are locally deposited. This can cause high doses to some voxels in air. It is mainly because the range of secondary particles in the air cavity is larger than the voxel size. Ignoring electron transport might impact dose calculation results for lung cancer patients. Previous literature has identified a similar problem (Jia *et al* 2012). We will include electron transport in air in future improvements of the code. Further, moqui currently only supports pencil beam scanning treatments because it relies on a beam model. We will implement additional functionalities, such as primary particle generation from phase space data and aperture components, to offer various treatment setups.

The hash-table implementation in moqui could be translated into a CPU implementation and the data structure is also provided with the standard library of C++.

Though we assumed a constant production rate of proton particles in this study, moqui could read Ion Beam Treatment Record with support from the DICOM-RT-Ion interface. Therefore, one could implement an appropriate parser for machine log files and their own machine-specific parameters to calculate the dose rate of actual delivery or dose rate in sub-spot level from it using moqui.

5. Conclusions

We developed a novel GPU-based MC code for proton dose calculation. The code transports protons considering multiple scattering, ionization, proton-proton elastic, proton-oxygen elastic, and proton-oxygen inelastic interactions. We employed the hash-table data structure, one of the key-value data structures, to improve the memory efficiency of the code. With the data structure, we can measure the full D_{ij} matrix and dose-rate of a patient using a GPU within 10 minutes. The developed code will be available to the public as open-source (https://github.com/mghro/moquimc) so that users can implement their own treatment machines, physic models, or scorers for proton dose calculation.

Acknowledgment

This work is supported by NCI R01 234210 “Fast Individualized Delivery Adaptation in Proton Therapy.”

The authors thank Enterprise Research Infrastructure & Services at Partners HealthCare for their in-depth support and for the provision of the research computing cluster.

References

Agostinelli S, Allison J, Amako K, Apostolakis J, Araujo H, Arce P, Asai M, Axen D, Banerjee S, Barrand G, Behner F, Bellagamba L, Boudreau J, Broglia L, Brunengo A, Burkhardt H, Chauvie S, Chuma J, Chytracek R, Cooperman G, Cosmo G, Degtyarenko P, Dell’Acqua A, Depaola G, Dietrich D, Enami R, Feliciello A, Ferguson C, Fesefeldt H, Folger G, Foppiano F, Forti A, Garelli S, Giani S, Giannitrapani R, Gibin D, Cadenas JJG, González I, Abril GG, Greeniaus G, Greiner W, Grichine V, Grossheim A, Guatelli S, Gumplinger P, Hamatsu R, Hashimoto K, Hasui H, Heikkinen A, Howard A, Ivanchenko V, Johnson A, Jones FW, Kallenbach J, Kanaya N, Kawabata M, Kawabata Y, Kawaguti M, Kelner S, Kent P, Kimura A, Kodama T, Kokoulin R, Kossov M, Kurashige H, Lamanna E, Lampén T, Lara V, Lefebure V, Lei F, Liendl M, Lockman W, Longo F, Magni S, Maire M, Medernach E, Minamimoto K, Freitas P M de, Morita Y, Murakami K, Nagamatu M, Nartallo R, Nieminen P, Nishimura T, Ohtsubo K, Okamura M, O'Neale S, Oohata Y, Paech K, Perl J, Pfeiffer A, Pia MG, Ranjard F, Rybin A, Sadilov S, Salvo ED, Santin G, Sasaki T, et al. 2003 Geant4—a simulation toolkit Nucl Instruments Methods Phys Res Sect Accel Spectrometers Detect Assoc Equip 506 250–303

Andreo P 2015 Dose to 'water-like' media or dose to tissue in MV photons radiotherapy treatment planning: still a matter of debate Phys Medicine Biology 60 309–37

Appleby A 2011 MurmurHash Online: https://sites.google.com/site/murmurhash/

Bobić M, Lalonde A, Sharp GC, Grassberger C, Verburg JM, Winey BA, Lomax AJ and Paganetti H 2021 Comparison of weekly and daily online adaptation for head and neck intensity-modulated proton therapy Phys Medicine Biology 66 055023

Cortés-Giraldo MA and Carabe A 2015 A critical study of different Monte Carlo scoring methods of dose average linear-energy-transfer maps calculated in voxelized geometries irradiated with clinical proton beams Phys Med Biol 60 2645–69 [PubMed: 25768028]

Farrell D 2020 A simple GPU hash table Online: https://github.com/nosferalatu/SimpleGPUHashTable

Fippel M and Soukup M 2004 A Monte Carlo dose calculation algorithm for proton therapy Med Phys 31 2263–73 [PubMed: 15377093]

Fracchiolla F, Engwall E, Janson M, Tamm F, Lorentini S, Fellin F, Bertolini M, Algranati C, Righetto R, Farace P, Amichetti M and Schwarz M 2021 Clinical validation of a GPU-based Monte Carlo dose engine of a commercial treatment planning system for pencil beam scanning proton therapy Phys Medica 88 226–34

Gajewski J, Garbacz M, Chang C-W, Czerska K, Durante M, Krah N, Krzempek K, Kopeć R, Lin L, Mojżeszek N, Patera V, Pawlik-Niedzwiecka M, Rinaldi I, Rydygier M, Pluta E, Scifoni E, Skrzypek A, Tommasino F, Schiavi A and Rucinski A 2021 Commissioning of GPU–Accelerated Monte Carlo Code FRED for Clinical Applications in Proton Therapy Aip Conf Proc 8 567300

Giantsoudi D, Grassberger C, Craft D, Niemierko A, Trofimov A and Paganetti H 2013 Linear Energy Transfer-Guided Optimization in Intensity Modulated Proton Therapy: Feasibility Study and Clinical Potential Int J Radiat Oncol Biology Phys 87 216–22

Giantsoudi D, Schuemann J, Jia X, Dowdell S, Jiang S and Paganetti H 2015 Validation of a GPU-based Monte Carlo code (gPMC) for proton radiation therapy: clinical cases study Phys Med Biol 60 2257–69 [PubMed: 25715661]

Granville DA and Sawakuchi GO 2015 Comparison of linear energy transfer scoring techniques in Monte Carlo simulations of proton beams Phys Med Biol 60 N283–91 [PubMed: 26147442]

Grassberger C, Trofimov A, Lomax A and Paganetti H 2011 Variations in Linear Energy Transfer Within Clinical Proton Therapy Fields and the Potential for Biological Treatment Planning Int J Radiat Oncol Biology Phys 80 1559–66

Jia X, Schümann J, Paganetti H and Jiang SB 2012 GPU-based fast Monte Carlo dose calculation for proton therapy Phys Med Biol 57 7783–97 [PubMed: 23128424]

Kohno R, Hotta K, Nishioka S, Matsubara K, Tansho R and Suzuki T 2011 Clinical implementation of a GPU-based simplified Monte Carlo method for a treatment planning system of proton beam therapy Phys Med Biol 56 N287–94 [PubMed: 22036894]

Kohno R, Sakae T, Takada Y, Matsumoto K, Matsuda H, Nohtomi A, Terunuma T and Tsunashima Y 2002 Simplified Monte Carlo Dose Calculation for Therapeutic Proton Beams Jpn J Appl Phys 41 L294

Li Y, Tian Z, Shi F, Song T, Wu Z, Liu Y, Jiang S and Jia X 2015 A new Monte Carlo-based treatment plan optimization approach for intensity modulated radiation therapy Phys Med Biol 60 2903–19 [PubMed: 25776792]

Li Y, Tian Z, Song T, Wu Z, Liu Y, Jiang S and Jia X 2016 A new approach to integrate GPU-based Monte Carlo simulation into inverse treatment plan optimization for proton therapy Phys Med Biol 62 289–305 [PubMed: 27991456]

Lin L, Huang S, Kang M, Hiltunen P, Vanderstraeten R, Lindberg J, Siljamaki S, Wareing T, Davis I, Barnett A, McGhee J, Simone CB, Solberg TD, McDonough JE and Ainsley C 2017 A benchmarking method to evaluate the accuracy of a commercial proton monte carlo pencil beam scanning treatment planning system J Appl Clin Med Phys 18 44–9

Luebke D 2008 CUDA: SCALABLE PARALLEL PROGRAMMING FOR HIGH-PERFORMANCE SCIENTIFIC COMPUTING 2008 5th Ieee Int Symposium Biomed Imaging Nano Macro 836–8

Ma J, Beltran C, Tseung HSWC and Herman MG 2014 A GPU-accelerated and Monte Carlo-based intensity modulated proton therapy optimization system Med Phys 41 121707 [PubMed: 25471954]

Maneval D, Bouchard H, Ozell B and Després P 2017 Efficiency improvement in proton dose calculations with an equivalent restricted stopping power formalism Phys Medicine Biology 63 015019

Maneval D, Ozell B and Després P 2019 pGPUMCD : an efficient GPU-based Monte Carlo code for accurate proton dose calculations Phys Medicine Biology 64 085018

Maurer WD and Lewis TG 1975 Hash Table Methods Acm Comput Surv Csur 7 5–19

Paganetti H 2009 Dose to water versus dose to medium in proton beam therapy Phys Med Biol 54 4399–421 [PubMed: 19550004]

Paganetti H 2012 Range uncertainties in proton therapy and the role of Monte Carlo simulations Phys Med Biol 57 R99–117 [PubMed: 22571913]

Pagh A, Pagh R and Rui M 2009 Linear Probing with Constant Independence Siam J Comput 39 1107–20

Pepin MD, Tryggestad E, Tseung HSWC, Johnson JE, Herman MG and Beltran C 2018 A Monte-Carlo-based and GPU-accelerated 4D-dose calculator for a pencil beam scanning proton therapy system Med Phys 45 5293–304 [PubMed: 30203550]

Perl J, Shin J, Schümann J, Faddegon B and Paganetti H 2012 TOPAS: An innovative proton Monte Carlo platform for research and clinical applications Med Phys 39 6818–37 [PubMed: 23127075]

Qin N, Botas P, Giantsoudi D, Schuemann J, Tian Z, Jiang SB, Paganetti H and Jia X 2016 Recent developments and comprehensive evaluations of a GPU-based Monte Carlo package for proton therapy Phys Med Biol 61 7347–62 [PubMed: 27694712]

Rogers DWO, Kawrakow I, Seuntjens JP, Walters BRB and Mainegra-Hing E 2021 NRC User Codes for EGSnrc NRC Report PIRS-702(revC)

Salvat F, Fernández-Varea JM and Sempau J 2009 PENELOPE-2008: A Code System for Monte Carlo Simulation of Electron and Photon Transport (Barcelona, Spain)

Schiavi A, Senzacqua M, Pioli S, Mairani A, Magro G, Molinelli S, Ciocca M, Battistoni G and Patera V 2017 Fred: a GPU-accelerated fast-Monte Carlo code for rapid treatment plan recalculation in ion beam therapy Phys Medicine Biology 62 7482–504

Schneider W, Bortfeld T and Schlegel W 2000 Correlation between CT numbers and tissue parameters needed for Monte Carlo simulations of clinical dose distributions Phys Med Biol 45 459–78 [PubMed: 10701515]

Shin J, Kooy HM, Paganetti H and Clasie B 2020 DICOM-RT Ion interface to utilize MC simulations in routine clinical workflow for proton pencil beam radiotherapy Phys Medica 74 1–10

Souris K, Lee JA and Sterpin E 2016 Fast multipurpose Monte Carlo simulation for proton therapy using multi- and many-core CPU architectures Med Phys 43 1700–12 [PubMed: 27036568]

Tseung HWC, Ma J and Beltran C 2015 A fast GPU-based Monte Carlo simulation of proton transport with detailed modeling of nonelastic interactions, Med Phys 42 2967–78 [PubMed: 26127050]

Unkelbach J, Botas P, Giantsoudi D, Gorissen BL and Paganetti H 2016 Reoptimization of Intensity Modulated Proton Therapy Plans Based on Linear Energy Transfer, Int J Radiat Oncol Biology Phys 96 1097–106

Unkelbach J, Zeng C and Engelsman M 2013 Simultaneous optimization of dose distributions and fractionation schemes in particle radiotherapy Med Phys 40 091702 [PubMed: 24007135]

Walters BRB, Kawrakow I and Rogers DWO 2002 History by history statistical estimators in the BEAM code system Med Phys 29 2745–52 [PubMed: 12512706]

Water S van de, Safai S, Schippers JM, Weber DC and Lomax AJ 2019 Towards FLASH proton therapy: the impact of treatment planning and machine characteristics on achievable dose rates Acta Oncol 58 1–7 [PubMed: 30698061]

Williams A, Barrus S, Morley RK and Shirley P 2005 An efficient and robust ray-box intersection algorithm Acm Siggraph 2005 Courses - Siggraph ‘05 9-es

Yepes PP, Mirkovic D and Taddei PJ 2010 A GPU implementation of a track-repeating algorithm for proton radiotherapy dose calculations Phys Med Biol 55 7107 [PubMed: 21076192]

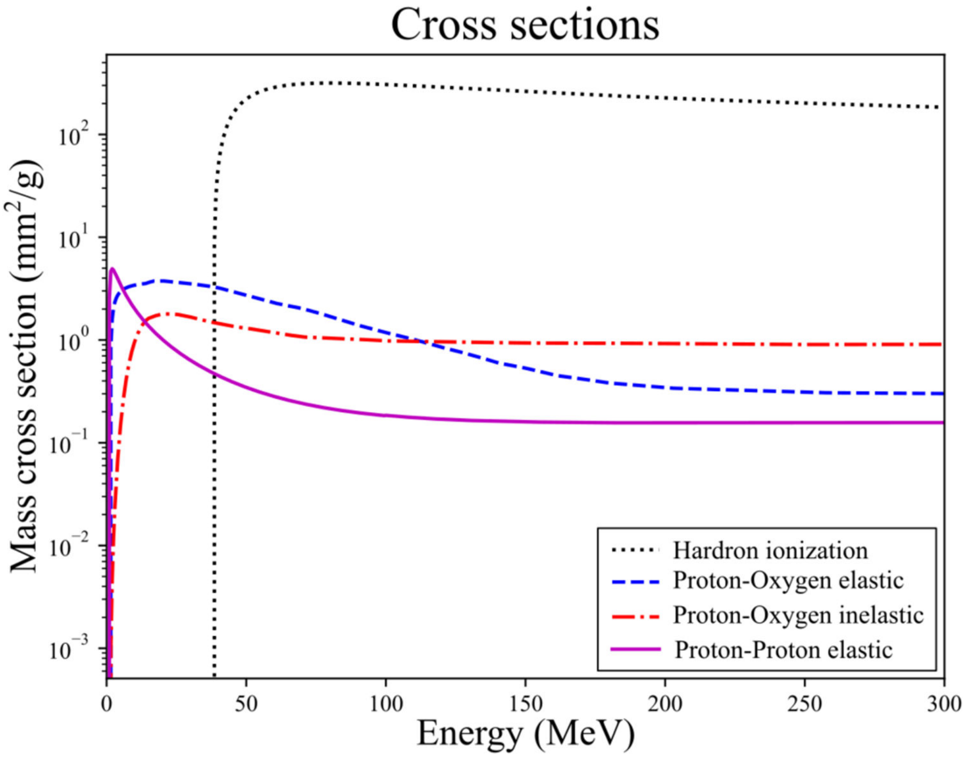


Figure 1.

Cross-section data of discrete interactions taken from Geant4

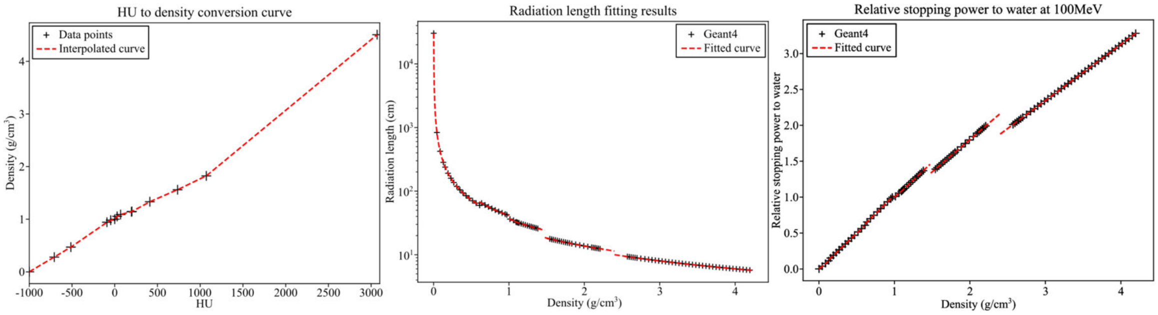


Figure 2.

Material data used in moqui. (left) HU to density conversion curve to select material from CT images, (middle) material density to radiation length conversion curve to determine scattering power, and (right) material density to relative stopping power to water conversion curve for 100 MeV proton.

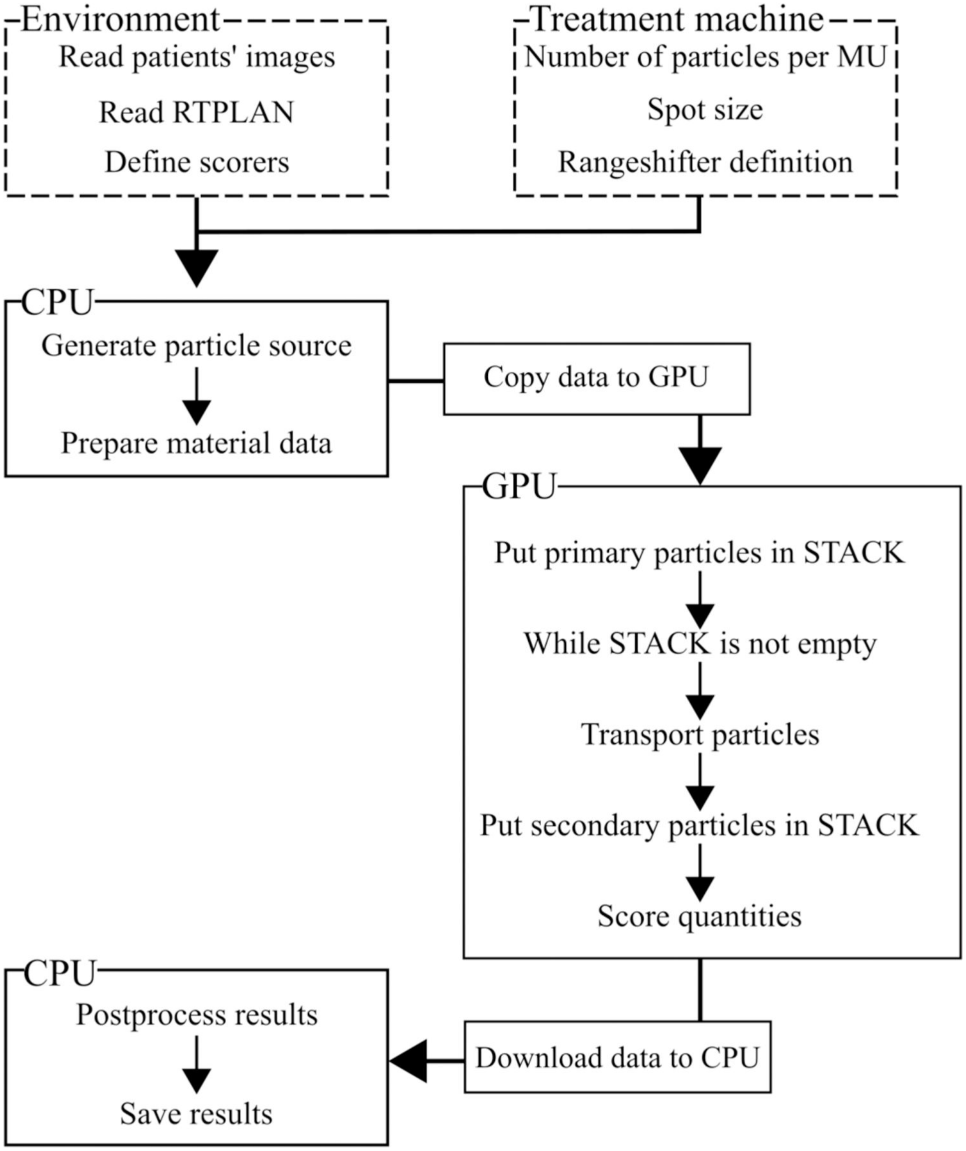


Figure 3.

Code structure of moqui

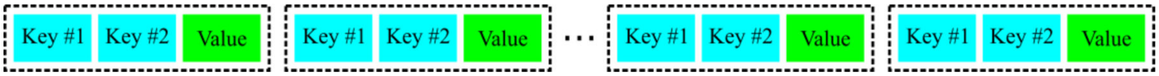
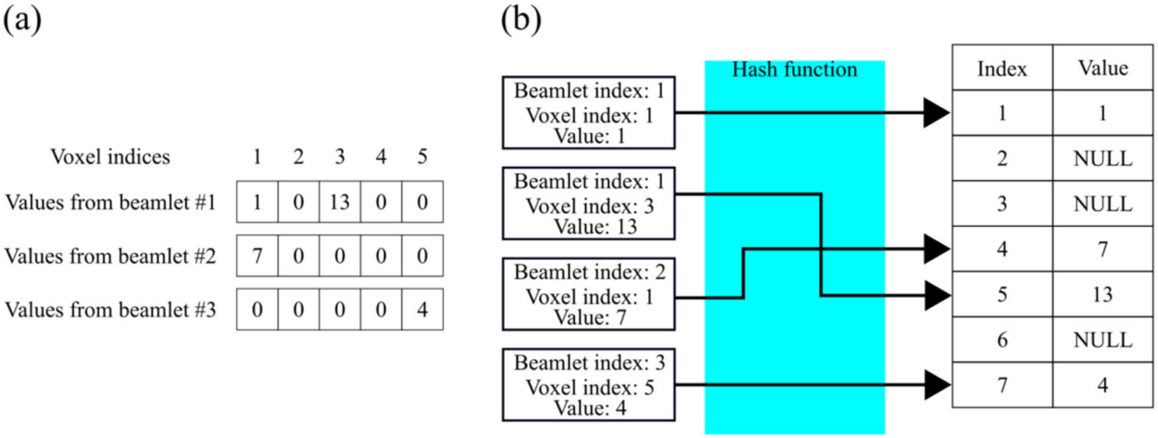


Figure 4.

Key-value data structure for moqui scoring system



Voxel indices	1	2	3	4	5
Values from beamlet #1	1	0	13	0	0
Values from beamlet #2	7	0	0	0	0
Values from beamlet #3	0	0	0	0	4

Index	Value
1	1
2	NULL
3	NULL
4	7
5	13
6	NULL
7	4

Figure 5.

Comparison between different data structures. (a) Array data structure and (b) hash-table data structure. One-to-one mapping between the scoring grid to the array data structure is required, while the hash-table only stores data for voxels that interact with the particle.

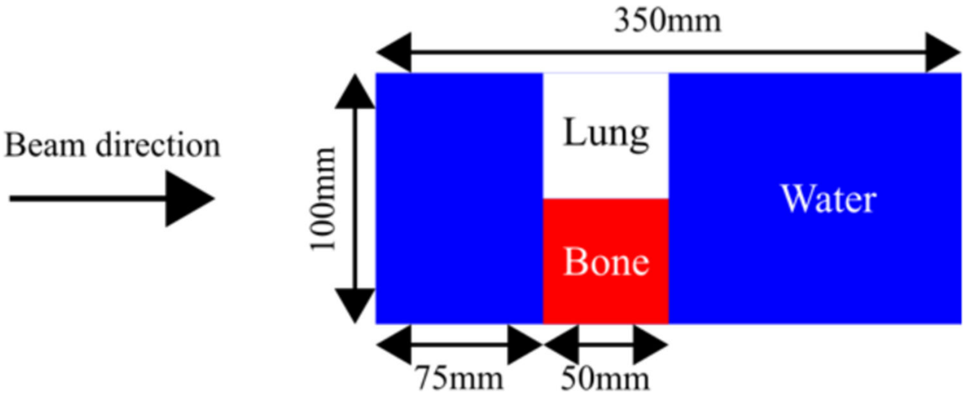


Figure 6.
Heterogeneous phantom

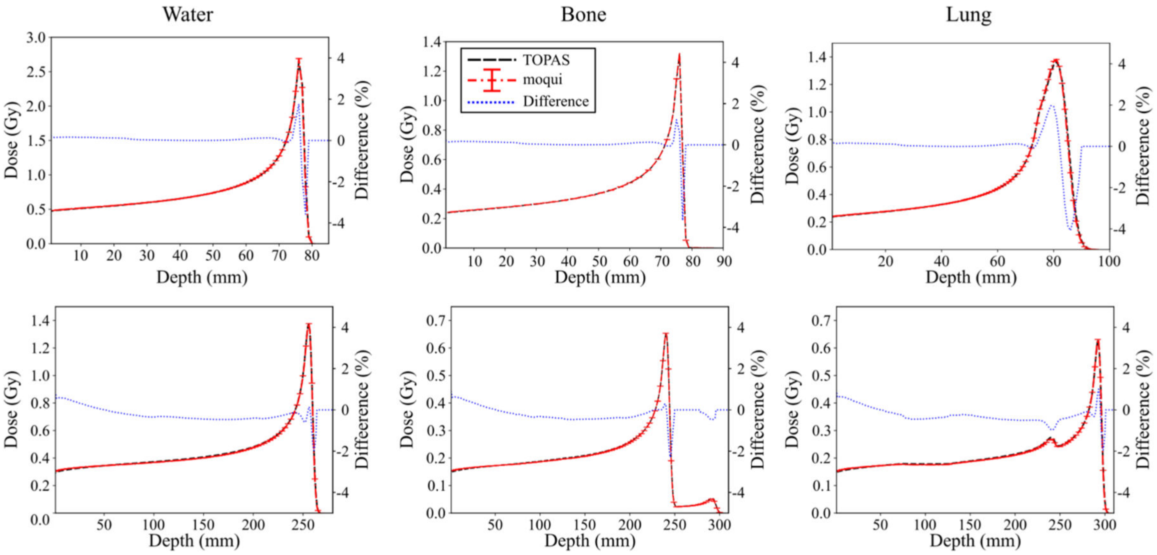


Figure 7.

Integrated-depth-dose in phantoms with primary proton energy of 100 MeV (top) and 200 MeV (bottom). Left: IDD curves in a homogeneous water phantom; Middle: IDD curves in the bone region of a inhomogeneous phantom; Right: IDD curves in the lung region of a inhomogeneous phantom.

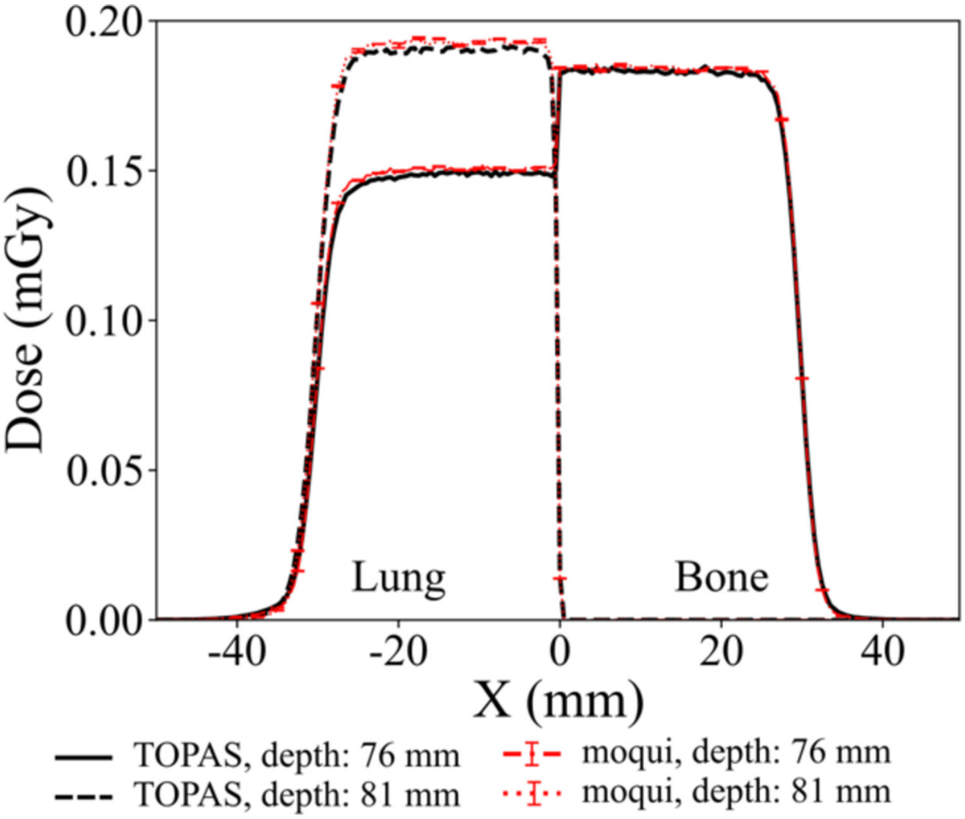


Figure 8.

Lateral profiles of dose deposition in the inhomogeneous phantom with 100 MeV primary protons at 76 mm and 81 mm from the entrance of the phantom.

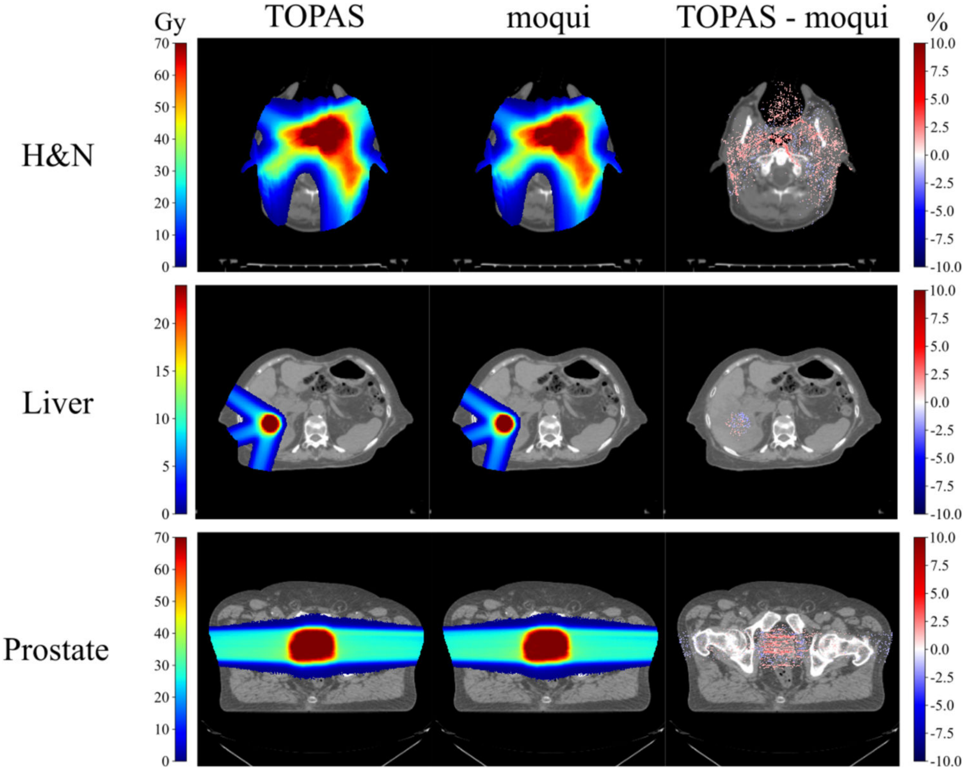


Figure 9.

Dose calculation results from different MC codes

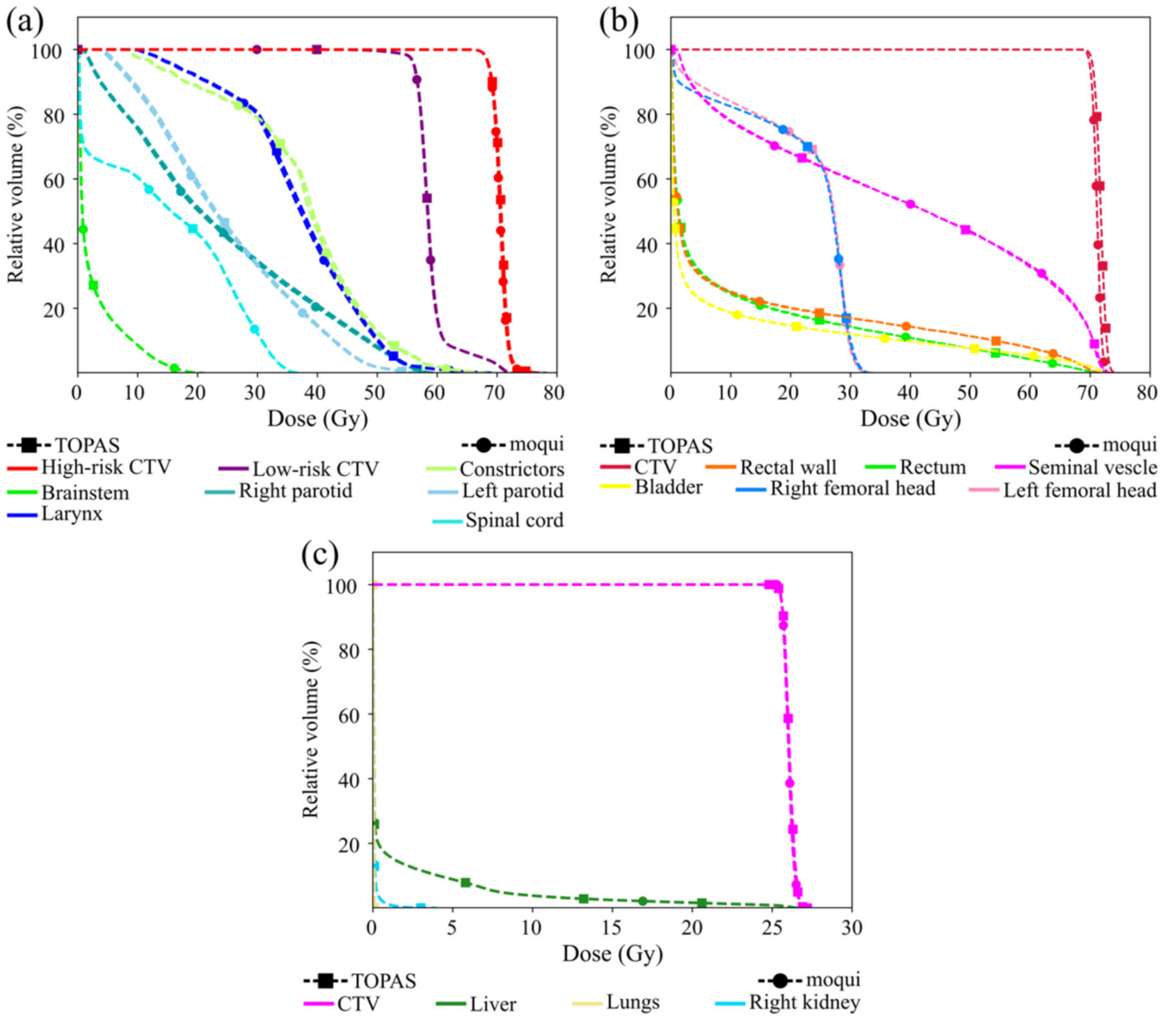


Figure 10.

Dose-volume histogram of the (a) H&N case, (b) prostate case, and (c) liver case

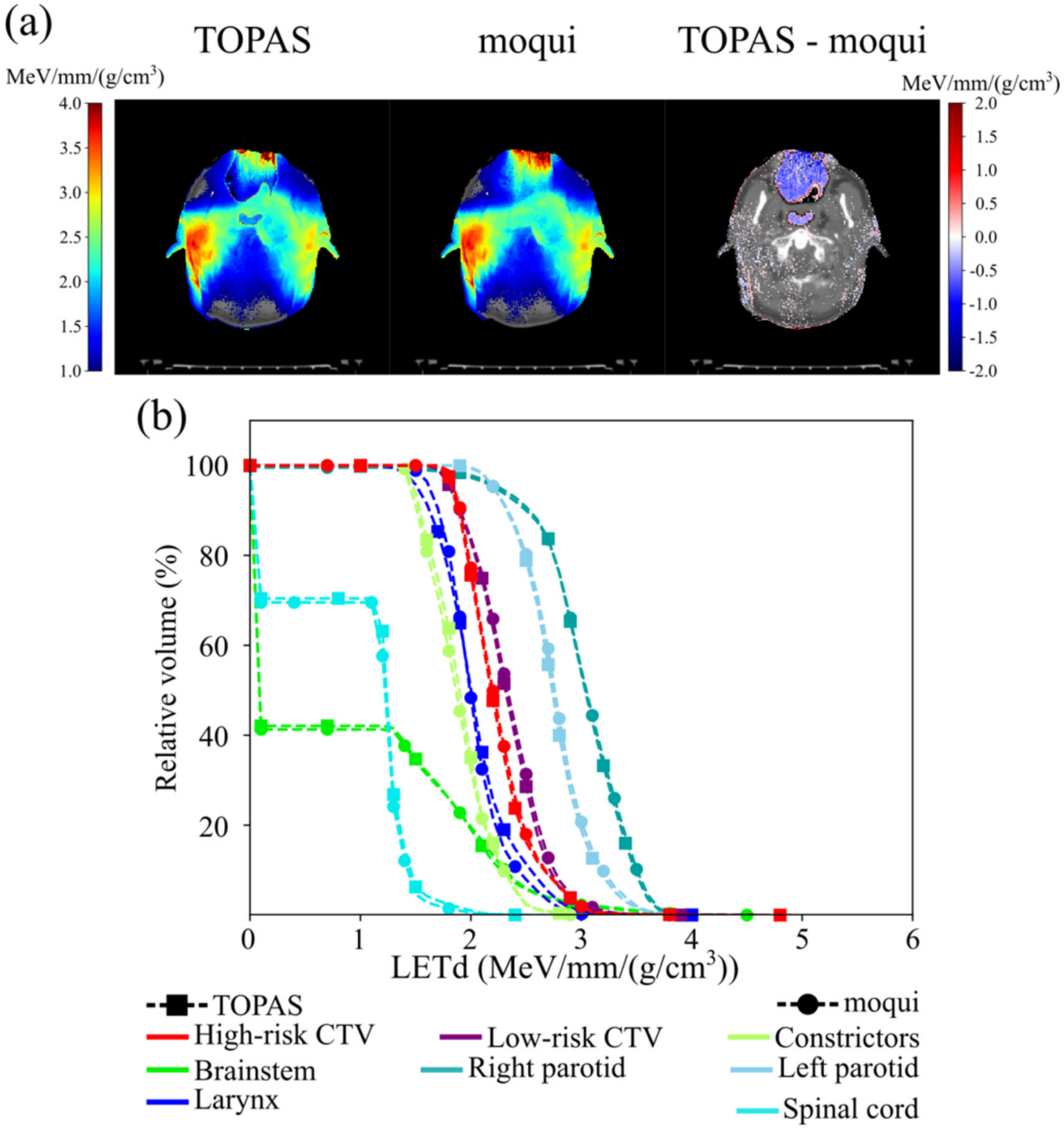


Figure 11.

(a) LET_d results from TOPAS, moqui, and their difference and (b) LET_d-volume histogram of the H&N case

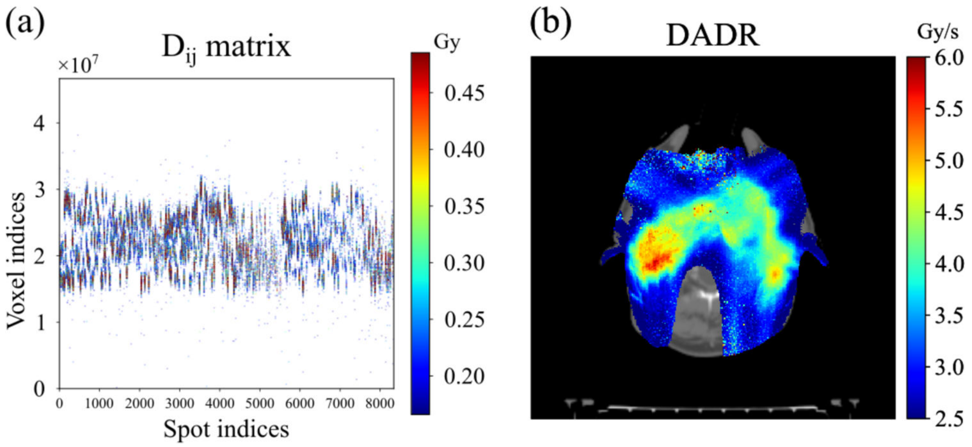


Figure 12.

D_{ij} matrix and DADR of the H&N case

Table I.

Treatment data of patients' cases

Case	Prescription dose (Gy)	Beam angles (degree)	Number of fractions	Range shifter thickness	The number of primary particles per batch
H&N	70 (High-risk CTV) 57 (Low-risk CTV)	180 (Beam #1) 60 (Beam #2) 300 (Beam #3)	34	35 mm	6,221,770 (Beam #1) 7,260,848 (Beam #2) 6,403,450 (Beam #3)
Liver	24	195 (Beam #1) 250 (Beam #2) 305 (Beam #3)	3	None	3,073,272 (Beam #1) 2,621,078 (Beam #2) 2,938,634 (Beam #3)
Prostate	70	90 (Beam #1) 270 (Beam #2)	28	None	6,290,841 (Beam #1) 6,147,573 (Beam #2)

Table II.

Gamma pass rate for the phantom cases

Phantoms	Energies	2mm/2%	0mm/2%	1mm/1%
Water phantom	50 MeV	100.0 %	72.11 %	99.81 %
	100 MeV	99.96 %	72.63 %	97.58 %
	150 MeV	99.94 %	74.34 %	97.10 %
	200 MeV	99.85 %	75.52 %	96.4 %
Inhomogeneous phantom	50 MeV	99.99 %	80.48 %	99.54 %
	100 MeV	99.99 %	79.59 %	99.04 %
	150 MeV	99.92 %	76.05 %	97.29 %
	200 MeV	99.78 %	71.55 %	95.90 %

Table III.

Gamma pass rate for patient cases

Case	2mm/2%	0mm/2%	1mm/1%
H&N	99.92	76.53	97.31
Liver	100 %	85.34	99.13
Prostate	99.34	73.43	95.67

Table IV.

Runtime benchmark of TOPAS and moqui

Cases	TOPAS (hours)	TOPAS – No secondary particles (hours)	Moqui (seconds)
H&N	11.85	7.97	31.87
Liver	7.36	3.02	27.03
Prostate	25.65	9.95	36.49