Development of a fast Monte Carlo dose calculation system for online adaptive radiation therapy quality assurance

2017 Phys. Med. Biol. 62 4970

(http://iopscience.iop.org/0031-9155/62/12/4970)

IOP Publishing | Institute of Physics and Engineering in Medicine Physics in Medicine & Biology

Phys. Med. Biol. **62** (2017) 4970–4990 https://doi.org/10.1088/1361-6560/aa6e38

Development of a fast Monte Carlo dose calculation system for online adaptive radiation therapy quality assurance

Yuhe Wang, Thomas R Mazur, Justin C Park, Deshan Yang, Sasa Mutic and H Harold Li

Department of Radiation Oncology, Washington University School of Medicine, St. Louis, MO 63110, United States of America

E-mail: hli@wustl.edu

Received 9 January 2017, revised 9 April 2017
Accepted for publication 20 April 2017
Published 22 May 2017

Abstract:
Online adaptive radiation therapy (ART) based on real-time magnetic resonance imaging represents a paradigm-changing treatment scheme. However, conventional quality assurance (QA) methods based on phantom measurements are not feasible with the patient on the treatment couch. The purpose of this work is to develop a fast Monte Carlo system for validating online re-optimized tri-^{60}Co IMRT adaptive plans with both high accuracy and speed. The Monte Carlo system is based on dose planning method (DPM) code with further simplification of electron transport and consideration of external magnetic fields. A vendor-provided head model was incorporated into the code. Both GPU acceleration and variance reduction were implemented. Additionally, to facilitate real-time decision support, a C++ GUI was developed for visualizing 3D dose distributions and performing various analyses in an online adaptive setting. A thoroughly validated Monte Carlo code (gPENELOPE) was used to benchmark the new system, named GPU-accelerated DPM with variance reduction (gDPMvr). The comparison using 15 clinical IMRT plans demonstrated that gDPMvr typically runs 43 times faster with only 0.5% loss in accuracy. Moreover, gDPMvr reached 1% local dose uncertainty within 2.3 min on average, and thus is well-suited for ART QA.

Keywords: GPU, Monte Carlo, variance reduction, MRI guided radiation therapy, adaptive radiation therapy

(Some figures may appear in colour only in the online journal)

1. Introduction

Online adaptive radiation therapy (ART) enables treatment adjustment based on on-board imaging immediately before treatment delivery to account for physical or functional changes to the target volume and organs at risk (Lagendijk *et al* 2014, Acharya *et al* 2016). However, with the patient on the treatment couch it is not feasible to validate the newly created intensity modulated radiation therapy (IMRT) plan using conventional patient-specific quality assurance (QA) approaches that rely heavily on comparison between planned dose and dose measured in a physical phantom. An alternative approach is performing a second dose calculation for the plan with an independently commissioned dose calculation engine to verify the dose distribution provided by the online treatment planning system (Noel *et al* 2014). A fast Monte Carlo platform for independently verifying dose distributions in MRI-guided ART is preferable to accurately simulate charged particle transport in external magnetic fields (Raaijmakers *et al* 2008). Venerable Monte Carlo simulation packages such as GEANT4 (Ahmad *et al* 2016), EGS4/EGSnrc (Malkov and Rogers 2016), and PENELOPE (Faddegon *et al* 2009) have been demonstrated to agree excellently with experimental data under a wide range of conditions, but these packages typically require many hours or even days to achieve an adequate statistical uncertainty (e.g. 1%), which is far beyond the time constraint imposed by an online adaptive scheme. Ideally we need a fast Monte Carlo engine that can complete a 3D dose calculation in a few minutes while maintaining sufficiently high accuracy.

Three approaches have been considered for improving Monte Carlo calculation efficiency (Chetty *et al* 2007) including: (1) simplifying particle transport mechanisms, thus reducing the necessary time for each particle history (Sempau *et al* 2000), (2) using variance reduction techniques such as particle splitting, Russian roulette, and interaction forcing to reduce the total history number required to achieve a given uncertainty (Kawrakow and Fippel 2000), and (3) enhancing computational capability by parallelizing the simulation with multiple CPU or GPU threads (Jia *et al* 2010, 2011, Hissoiny *et al* 2011, Jahnke *et al* 2012). Approaches (1) and (2) alter the physical mechanisms of particle transport, and thus may compromise accuracy.

To facilitate computational dosimetry for the MRIdian system (ViewRay, Inc., Cleveland, OH), the only MRI-guided radiotherapy (MRgRT) system currently in clinical use, we recently developed and experimentally validated a GPU-accelerated Monte Carlo dose calculation package called gPENELOPE based on PENELOPE that employs approach (3) only (Wang *et al* 2016, Rankine *et al* 2017). This package can simulate a tri-^{60}Co IMRT plan subject to a 0.35 T magnetic field in about 1 h with less than 1% average local uncertainty in a volume where the dose is greater than 10% of the dose maximum (Wang *et al* 2016). While substantially faster than PENELOPE, gPENELOPE is still not fast enough for online adaptive plan verification.

An alternative fast Monte Carlo code is the dose planning method (DPM) (Sempau *et al* 2000) that employs a simplified coupled electron–photon transport scheme in order to achieve high computational performance. More recently, DPM was accelerated via a GPU implementation, gDPM (Jia *et al* 2010). According to our benchmarks, the mean time for calculating dose to $<1\%$ local uncertainty is 13.8 min using gDPM for 15 clinical MRIdian IMRT plans (see section 3). While substantially faster than gPENELOPE, the median time for online adaptation measured at our institution to date is 26 min (Acharya *et al* 2016). A faster dose calculation platform—likely with additional simplifications—is thus required for consideration in our ART protocol.

In this study, we incorporate the vendor-provided MRIdian head model into DPM and allow for consideration of magnetic fields. We then accelerate the code via GPU implementation, yielding an MRIdian-specific version of gDPM (referred to as gDPM for simplicity in

this manuscript). In addition to GPU acceleration (i.e. gDPM), we further accelerate gDPM by introducing (1) variance reduction techniques and (2) additional physical simplifications enabled by details of the MRIdian platform to enable fast and accurate Monte Carlo dose calculation for online ART. We present detailed comparisons of the resulting code, GPU-accelerated DPM with variance reduction (gDPMvr), against gPENELOPE and gDPM in a variety of phantoms to demonstrate that gDPMvr achieves the required calculation efficiency for ART while maintaining sufficient accuracy to engender physicists' confidence in adaptive plan QA.

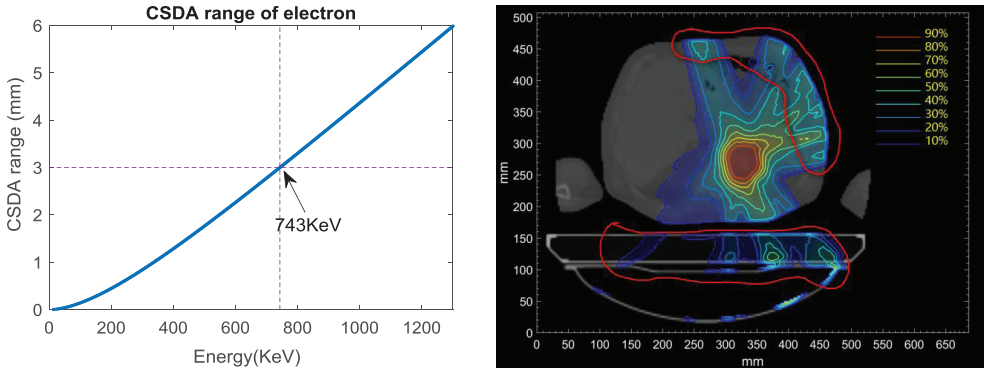
2. Methods and materials

We first analyze the key differences between DPM and PENELOPE, and then suggest additional simplifications to DPM based on the MRIdian system's features. Then we apply the particle splitting and Russian roulette variance reduction algorithms (Kawrakow and Fippel 2000) together with approximations for particle transport in a magnetic field to build a GPU-accelerated workflow. We compare the performance of gPENELOPE, gDPM, and gDPMvr in terms of speed and accuracy using a variety of digital phantoms and clinical plans.

2.1. gPENELOPE versus gDPM on the MRIdian system

Compared to the general-purpose package PENELOPE, DPM includes a number of simplifications and optimizations for dose calculation in a patient (Sempau *et al* 2000). (1) DPM covers a narrower range of energy so all relevant cross-sections can be obtained using spline or linear interpolation instead of having to perform interpolation on a logarithmic scale, thus enabling faster data access. (2) DPM ignores several types of interaction with either low probability of occurrence in the energy range of interest or little impact on the final dose, such as Rayleigh scattering and inner shell ionization. (3) DPM modifies the pair-production mechanism by tracking the positron as an electron and emitting two annihilation photons in randomly selected opposite directions to compensate for the latent energy from the positron. This simplification reduces the data table and code length by almost a third. (4) DPM uses simpler physics models to describe scattering events. Taking Compton scattering as an example, DPM applies the Klein–Nishina formula which treats the electrons as free and at rest despite the fact that electrons are bound to atoms in a shell structure with specific energies. (5) DPM uses a random energy hinge for multiple electron scattering instead of a random step hinge used in PENELOPE. This modification enables larger path lengths in multiple scattering events, thereby improving simulation efficiency.

Like PENELOPE, DPM uses a 'mixed' scheme for electron transport by treating large energy transfer collisions in an analogue sense and using the continuous slowing down approximation (CSDA) to model small-loss collisions. For online adaptive treatments on the MRIdian system, two facts can be exploited to simplify this scheme. First, ^{60}Co emits two gamma rays with energies of just 1.17 and 1.33 MeV, much lower than the photon energy generated by a typical LINAC. Second, our clinic uses a voxel size of $3 \times 3 \times 3\,\mathrm{mm}^3$ for treatment planning. Figure 1(a) shows how an electron's CSDA range varies with energy in water. For gamma rays emitted by ^{60}Co decay, secondary electrons will never travel a distance of more than two voxels. Our simulation profiles with gPENELOPE also show that approximately 80% of secondary electrons have energy less than 743 keV, i.e. the threshold energy for travelling one voxel distance. The remaining secondary electrons are mostly generated around the beam entrance, which is usually far away from the target region as shown in figure 1(b). In other words, most electrons will exhaust all their energy in their voxel-of-origin and neighboring



(a) (b)

Figure 1. (a) CSDA range of an electron versus kinetic energy in water. (b) Dose distribution calculated for a clinical IMRT plan. The isodose lines are shown relative to the maximum dose. The area enclosed by the red lines indicate the area of energy deposition by photons with energies greater than 743 KeV.

voxel, and so detailed analogue simulation is hardly necessary for determining in which voxel an electron deposits its energy. We therefore can just apply the CSDA approximation in order to greatly simplify the code at the expense of minor accuracy loss. Since the CSDA implementation is simple and fast, we can process secondary electrons immediately in each photon event, instead of storing them in stacks for subsequent processing. Likewise, the 1.33 MeV maximum photon energy allows for a single pair-production event at most per history, and so similarly no stack structure is required. Eliminating stack requirements mitigates the thread divergence phenomenon on GPUs, and hence improves execution efficiency.

In summary, Compton scattering is modeled via the Klein–Nishina equation, pair production is modeled by sampling energy uniformly between 0 and $E_{\mathrm{P}} - 2E_{\mathrm{S}}$ where E_{P} is the photon energy and E_{S} the electron rest mass energy, and the photoelectric effect is modeled by simply changing the particle property label from photon to electron. Woodcock tracking (Woodcock *et al* 1965) is applied to handle photon transport in a heterogeneous phantom effectively. As mentioned above, only the CSDA scheme is used to handle electron transport with changes in direction being derived from a random energy hinge. A fixed energy loss segment E_d (e.g. 200 KeV) is split in two sub-steps (Sempau *et al* 2000)

$$E_A = \xi E_d \text{ and } E_B = E_d - E_A, \tag{1}$$

where ξ is a random number distributed uniformly between 0 and 1. The electron will first advance a certain distance that exhausts E_A energy along the initial direction, then get deflected by multiple-scattering and advance another distance that exhausts E_B along the new direction. If the electron's initial energy $E_e < E_d$, then E_d is replaced by E_e in the above equation.

2.2. gDPM with variance reduction

To further improve computational efficiency, we applied particle splitting and Russian roulette variance reduction methods proposed by Kawrakow *et al* (2000), which can significantly reduce the necessary number of photon scattering events. Suppose we split the incident photon into N_S photons. The sampled distance to the interaction site of the ith photon can then be set as

$$s_i = -\lambda \log\left(1 - \frac{\xi + i}{N_S}\right), \tag{2}$$

where ξ is a uniform random number between 0 and 1 and λ is the mean free path of photon. This equation distributes N_S photons along the initial trajectory according to a log distribution at the expense of a single random number ξ. These photons undergo the same interaction with the weight assigned to produced secondary particles reduced to $1/N_S$. Only one primary photon is randomly selected to continue its history and its weight is recovered to 1. The idea of this method is to generate N_S electrons spread along a path for one photon interaction, saving approximately $(N_S - 1)/N_S$ of the photon simulation time. Since the cost of photon scattering is more expensive than the concise CSDA model of electron transport, this method significantly improves overall efficiency. Although the approximation inappropriately assumes that secondary electrons at these sites share the same energy and direction, the defect is blurred by a large history number of random scatterings in the subsequent electron transport. Benchmarks confirm that these variance reduction methods can effectively improve the calculation speed at the expense of minor loss in accuracy (see section 3).

2.3. Transport in magnetic field

Given that the CSDA range $\bar{s}$ of electrons and positrons is typically very small, the magnetic field in each voxel can be treated as uniform in most applications. The particles will undergo spiral motion in a uniform magnetic field $\boldsymbol{B}$ at the relativistic angular velocity

$$\vec{\omega} = -\frac{e\vec{B}}{\gamma m_e}, \tag{3}$$

where e denotes the elementary charge, m_e is the electron mass, and γ is the Lorentz factor. The corresponding location after advancing a length s in a uniform phantom can be readily evaluated as shown in the PENELOPE user manual to be

$$\vec{r}(s) = \vec{r}_0 + s\hat{v}_0 - \frac{s}{v_0}\vec{v}_{0\perp} + \frac{1}{\omega}[1 - \cos(s\omega/v_0)](\hat{\omega} \times \vec{v}_{0\perp}) + \frac{1}{\omega}\sin(s\omega/v_0)\vec{v}_{0\perp}, \tag{4}$$

where $\boldsymbol{r_0}$ is the initial particle location and $\boldsymbol{v_0}$ is the particle velocity (with $\boldsymbol{v_{0\perp}}$ being the velocity component perpendicular to $\boldsymbol{B}$). For a heterogeneous voxelized phantom, however, the intersection between the spiral curve and the voxel boundary must be calculated due to variation of the density in each voxel. Accurate evaluation can be inefficient due to many inverse trigonometric function calls. As the CSDA range $\bar{s}$ is relatively small in comparison to the spiral radius R, we can approximate the spiral motion by small straight line segments that change direction gradually (first-order Euler method). Allowing the error in one segment move to be Δ_{max}, the maximum segment length s_m is expressed as

$$s_m = \sqrt{(R + \Delta_{max})^2 - R^2} \approx \sqrt{2R\Delta_{max}}, \tag{5}$$

If $s > s_m$, we only advance a distance s_m and change direction by angle $\theta \approx s_m/R$ (continuing until s is exhausted). When moving a distance s crosses a voxel boundary, the particle direction may point back to the original voxel when $\boldsymbol{v} \cdot (\boldsymbol{v} + \mathrm{d}\boldsymbol{v}) < 0$, and so the current voxel index must be monitored accordingly.

2.4. GPU implementation

In contrast to CPU architecture, the GPU was originally designed for parallel graphic processing where single float precision is sufficient. On modern GPUs, single precision float

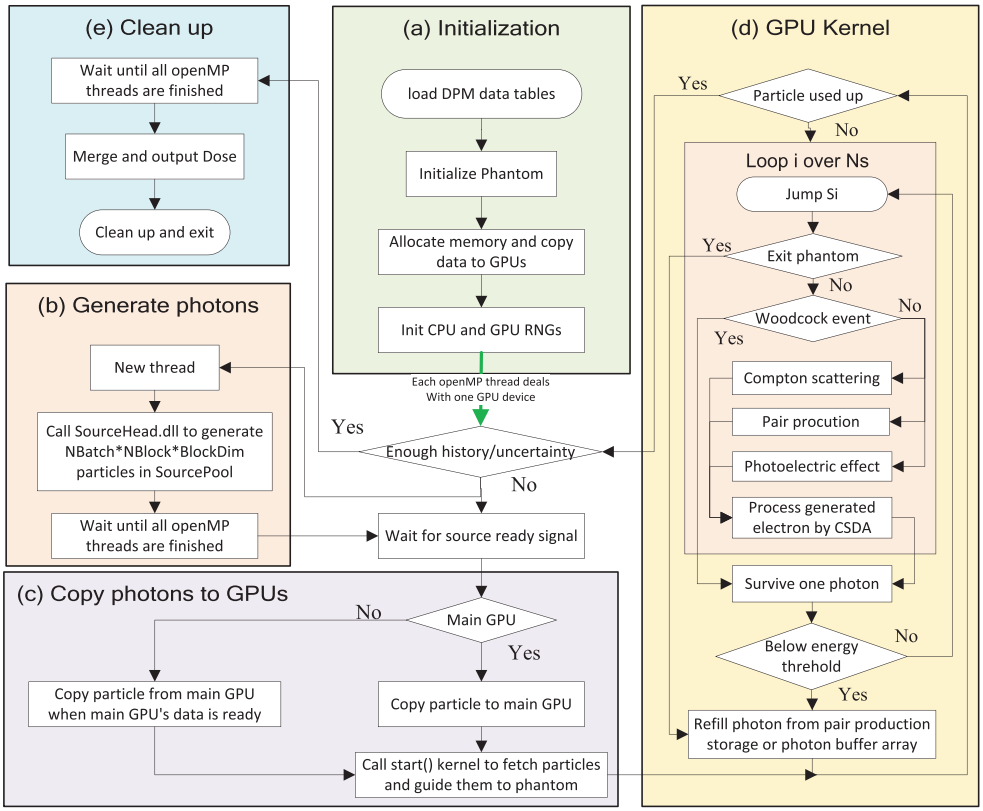


Figure 2. Workflow of gDPMvr including 5 modules: (a) initialization, (b) generate photons, (c) copy photons to GPUs, (d) GPU kernel, and (e) clean up.

operation is usually 2–3 times faster than double precision. To maximize performance, we decided to use single precision float numbers throughout our GPU code. Resulting accuracy loss has been proven to be negligible (Jia *et al* 2011).

In the original DPM implementation, spline interpolation is employed to calculate cross-sections. Though more accurate, spline interpolation costs four times the memory required by linear interpolation. In GPU architecture, device memory is large but also has a long accessing latency since it is only cached by a small L2 cache. Shared and constant memory are hundreds of times faster than device memory but have limited sizes. As shared memory cannot persist across different thread blocks, the best choice for cross-section tables is constant memory whose size is unfortunately only 64 KB for most Nvidia GPU devices. If spline interpolation were to be used, data tables would need to be loaded to device memory, and performance would be seriously compromised. Therefore, we decided to employ linear interpolation to shrink the data table size and thus utilize constant memory. Our simulation results presented later show marginal effects on final dose.

The workflow of our gDPMvr implementation is shown in figure 2. The program first reads and parses the configuration file which includes details regarding the GPU devices, phantom (geometry and materials), and source head. It then loads the DPM data tables, creates a digital phantom, allocates GPU memories and copies data tables to GPU devices as shown in figure 2(a). It also initializes the random number generator for each thread with a unique

seed. The K80 GPU card (Nvidia), on which we develop and test gDPMvr, includes two GPU devices. To enable multiple GPU support, our program launches multiple threads through OpenMP to call GPU kernel functions on each device simultaneously.

Meanwhile, the main thread launches another thread calling the vender-provided head source module to prepare one batch of incident photons as shown in figure 2(b). As specified in figure 2(c), the photons are first copied to the main GPU and then transferred to other GPUs to save I/O time. The GPU kernel function start() is called to guide photons to the phantom via free propagation, a reasonable approximation in air. The simulation kernel thread, as shown in figure 2(d), will split one incident photon N_S times in a loop, with each photon jumping distance s_i given by equation (2).

If not exiting the phantom, the program will test whether a Woodcock virtual event is selected. If yes, no scattering will happen. Otherwise, one event among Compton scattering, pair-production and photoelectric interaction will be selected based on their cross sections. Meanwhile, one or two electrons will be generated and then processed by using the CSDA scheme immediately. Since our electron transport model is fairly simple, it will not cause serious thread divergence on GPUs and thus it is not necessary to separate photon and electron transport as in the scheme used by gDPM (requiring N_S times bigger stacks and causing heavy data access with variance reduction).

If the energy of the surviving photon is above a specified cut-off energy, we repeat the splitting process. Otherwise, we refill the current particle variable either from pair-production storage or the incident photon buffer array, thus improving efficiency by enabling constant renewal of particles on all threads. We collect the dose and calculate the accumulated uncertainty after each batch is finished. When a given number of histories is finished or a specified uncertainty is reached, we merge the dose on each GPU device and free allocated memories as shown in figure 2(e).

Incident photons are generated by a vendor provided DLL module running on a CPU. In certain cases, the head model lags the GPU and cannot provide new particles at a sufficient rate. Our code provides an option to automatically balance the GPU and CPU workload by reusing certain photons. Simulation comparisons show that the dose differences caused by reusing source particles in 'hot areas' ($D > 10\% D_{\max}$) are mostly (99.73%) within the targeted 1% uncertainty for a large (10^9) history number (Wang *et al* 2016).

2.5. DoseViewer GUI

To simplify user operation in an online adaptive setting (Noel *et al* 2014), we developed a graphic user interface named DoseViewer to wrap the command line simulation kernel and provide visualization tools for 3D dose distributions as exemplified in figure 3. DoseViewer is based on a cross-platform C++ GUI library called *wxwidgets* and uses openGL to render 3D images. The interface has similar functionalities to software like CERR and 3D Slicer, such as viewing DICOM CT images, dose, beams, and contours in convenient ways, but responds much faster (programmed in C++) than CERR and has more compact size (about 10 MB) than 3D Slicer.

The biggest advantage of our in-house software, is its flexible customization. For example, this program can export a subset of the dose by visually clicking and dragging, export a slice of DICOM dose oriented at an arbitrary direction, and observe dose accumulating as a function of time. In addition, we developed a GPU-accelerated gamma analysis module to boost the performance of large matrix comparisons. This module is at least 4 times faster than CPU code that we tested (GT650M versus i7 3630QM). Lastly, DoseViewer also includes a multi-task managing module to launch simulations in batches.

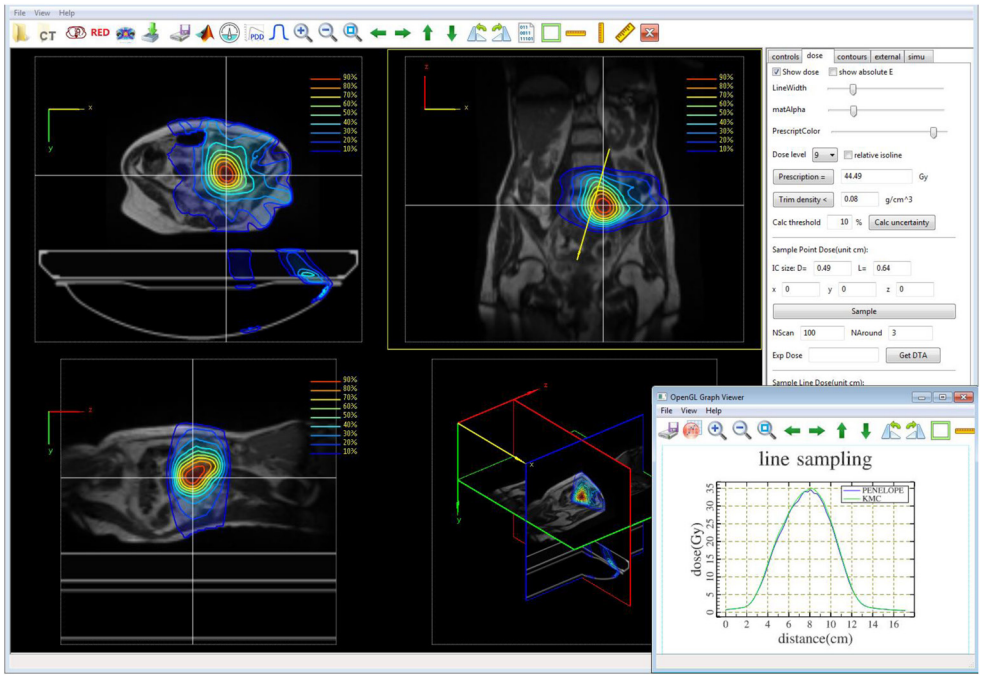


Figure 3. DoseViewer graphical user interface. The GUI has similar functionalities to CERR and 3D slicer, while providing more customized tasks with faster response and more compact size.

2.6. Methods of validating monte carlo systems

2.6.1. Phantoms. As shown in figure 4, four different types of phantoms (Ahmad *et al* 2016) were used to compare the accuracy and performance of the two DPM-based codes, gDPM and gDPMvr, to gPENELOPE. The first three phantoms are all synthetic phantoms sharing the same dimensions of $30.3 \times 30.3 \times 30.3\,\mathrm{cm}^3$. The first is a homogeneous water phantom and the second is a slab of uniform lung in water whose density is set to $0.3\,\mathrm{g\ cm}^{-3}$ according to PENELOPE's material database. The third phantom additionally includes a uniform tumor cube ($2.1 \times 2.1 \times 2.1\,\mathrm{cm}^3$) with a density of $0.7\,\mathrm{g\ cm}^{-3}$. In addition, we exported 15 patients' planning CT data from MRIdian's treatment planning system, with sites including stomach (4), lung (2), liver (3), adrenal gland (2), pancreas (2), spleen (1), and mediastinum (1). Calculating dose using the four types of objects provides a comprehensive evaluation of how well these algorithms perform in uniform, partially heterogeneous, and highly heterogeneous objects irradiated simultaneously by three ^{60}Co sources subject to a 0.35 T magnetic field. The voxel size in all cases is set to $3 \times 3 \times 3\,\mathrm{mm}^3$, the same as that we use in the clinic on MRIdian.

2.6.2. 3D dose comparisons. For 3D comparisons, we use both gamma passing rates and statistical histograms to reveal differences between Monte Carlo systems, i.e. gDPM, gDPMvr, and gPENELOPE. The gamma index for each voxel $\vec{r}$ is defined by Low *et al* (1998)

$$\gamma(\vec{r}) = \min\left\{\Gamma(\vec{r}, \vec{r}\,')\right\} \forall \left\{\vec{r}'\right\},$$
$$\Gamma(\vec{r}, \vec{r}\,') = \sqrt{\frac{|\vec{r} - \vec{r}\,'|^2}{\Delta d^2} + \frac{\left(D(\vec{r}) - D'(\vec{r}\,')\right)^2}{\Delta D^2}}, \tag{6}$$

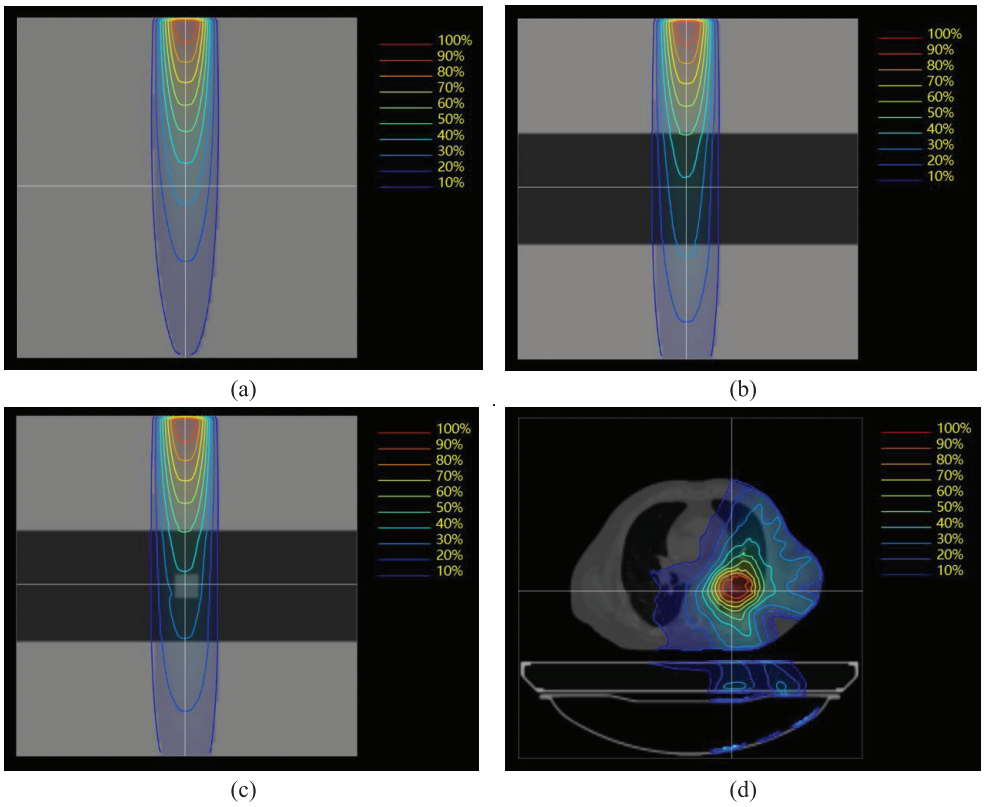


Figure 4. Phantoms used to evaluate accuracy and performance of gDPM, gDPMvr, and gPENELOPE. (a) $30.3 \times 30.3 \times 30.3\,\mathrm{cm}^3$ uniform water phantom, (b) water-lung-water phantom where lung's height is 9.9 cm, (c) water-lung-tumor-water phantom where tumor size is $2.1 \times 2.1 \times 2.1\,\mathrm{cm}^3$, and (d) patient. Voxel sizes are all set to $3 \times 3 \times 3\,\mathrm{mm}^3$.

where $|\vec{r} - \vec{r}'|$ represents the distance between voxels $\vec{r}$ and $\vec{r}'$, Δd is the distance-to-agreement (DTA) value, and ΔD is the dose tolerance value. We label a gamma index at voxel $\vec{r}$ as passing if $\gamma(\vec{r}) \leqslant 1.0$, and count the passing rate for those voxels where $D(\vec{r}) > t \cdot D_{\max}$, where t is a dose threshold. Higher gamma passing rates for smaller Δd and ΔD tolerances usually suggest stronger agreement between two dose distributions. Here we use strict criteria (DTA $= 0$, i.e. grid-to-grid comparison, $\Delta D = 0.5\% D_{\max}$, threshold $D(\vec{r}) > 10\% D_{\max}$) to amplify the differences as the three Monte Carlo engines behave quite similarly to each other.

A statistical histogram, on the other hand, visually indicates the distribution of dose differences spanning all voxels. Here we define a statistical variable z-score for each voxel as

$$\mathrm{z}(\vec{r}) = \frac{D_{\mathrm{test}}(\vec{r}) - D_{\mathrm{ref}}(\vec{r})}{D_{\mathrm{ref}}(\vec{r})} \cdot \frac{1}{\sigma_{\mathrm{tot}}} \tag{7}$$

where $D_{\mathrm{test}}(\vec{r})$ and $D_{\mathrm{ref}}(\vec{r})$ are test and reference dose at voxel $\vec{r}$ respectively, and σ_{tot} is the standard deviation of the distribution $(D_{\mathrm{test}}(\vec{r}) - D_{\mathrm{ref}}(\vec{r}))/D_{\mathrm{ref}}(\vec{r})$ spanning all voxels, which functions as a normalization factor and an indicator of the difference level. If two engines are identical, the histogram of z-scores will be a standard Gaussian distribution, and σ_{tot} will

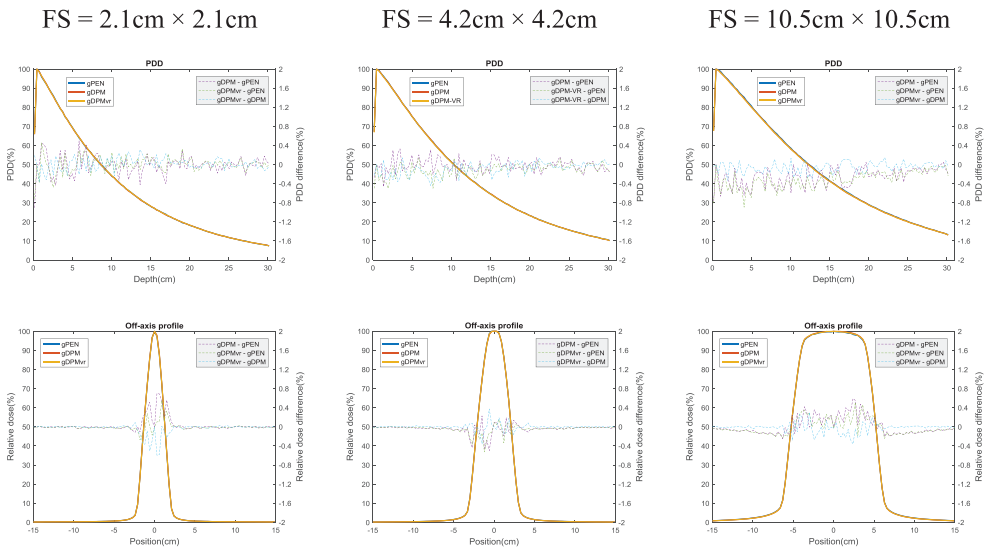


Figure 5. Percentage depth dose (upper row) and off-axis profiles (lower row, 5 cm depth) for a homogeneous water phantom. FS: field size.

approach $\sqrt{\sigma_{\text{test}}^2 + \sigma_{\text{ref}}^2}$, where σ_{test} and σ_{ref} are the uncertainties achieved by the two algorithms in question.

Aside from accuracy, we also performed detailed efficiency comparisons among the three Monte Carlo engines. Since the uncertainty σ is approximately proportional to $1/\sqrt{N}$ where N is the history number which is in turn proportional to simulation time T, i.e. $\sigma \cong 1/\sqrt{\eta T}$, we define this constant of proportionality η as the calculation efficiency:

$$\eta \cong \frac{1}{\sigma^2 T} \tag{8}$$

As the dose uncertainty σ_i varies by voxel, σ is calculated by squared root averaging, i.e.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} \sigma_i^2}{N}} \tag{9}$$

where N is the number of voxels whose dose is above a given threshold ($D(\vec{r}) > 10\% D_{\max}$ in our tests).

3. Results

Our previously developed dose calculation system gPENELOPE (Wang *et al* 2016) has been validated to be as accurate as the original PENELOPE code with significantly improved efficiency. We thus consider gPENELOPE as the standard throughout these comparisons. To evaluate efficiency and accuracy changes introduced by the variance reduction (rarely used in GPU code), we decided to cross-compare the results of gDPM and gDPMvr as well. For short we denote gPENELOPE as gPEN, the dose differences $D(\text{gDPM}) - D(\text{gPEN})$ as err1, $D(\text{gDPMvr}) - D(\text{gPEN})$ as err2, and $D(\text{gDPMvr}) - D(\text{gDPM})$ as err3 in the following figures. All simulations are performed on a single K80 GPU card (including two units) produced by Nvidia.

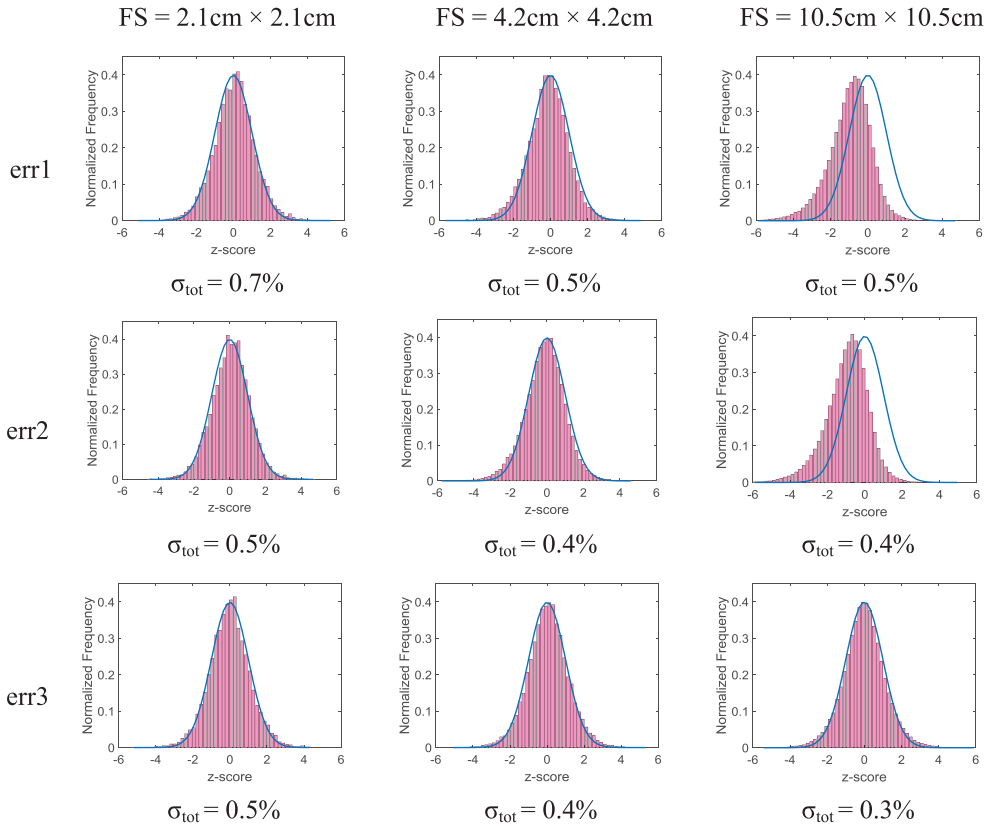


Figure 6. z-score histograms among gPEN, gDPM and gDPMvr for a homogeneous water uniform phantom. FS: field size.

3.1. Homogeneous water phantom

Figure 5 compares vertical and lateral dose profiles in a homogeneous water phantom for three field sizes. All the profiles from the three codes agree with each other to within 0.8%. Figure 6 compares the histograms of z-scores. For field sizes of $2.1 \times 2.1\,\mathrm{cm}^2$ and $4.2 \times 4.2\,\mathrm{cm}^2$, the histograms for all three algorithms are close to Gaussian. For a field size of $10 \times 10\,\mathrm{cm}^2$, systematic differences are observed: gDPM and gDPMvr score less dose than gPEN, which may be a result of the different ways of handling below-threshold energy photons in DPM (ignore) and PENELOPE (score). For larger field sizes, more energy (with a squared growth rate) is deposited so the difference becomes appreciable. We note that the histograms are normalized by σ_{tot}, so being observable does not imply large absolute differences. In fact the standard deviation of the $10 \times 10\,\mathrm{cm}^2$ field is actually smaller than that of smaller field sizes. The z-score histograms between gDPM and gDPMvr are always close to Gaussian, indicating that the introduction of variance reduction has negligible effect on accuracy.

Table 1 summarizes the performances achieved by the three algorithms. Both gPEN and gDPM ran 10^9 histories to achieve less than 0.5% uncertainty, while gDPMvr ran 10^8 histories to reach an even smaller uncertainty. The mean efficiency ratio, gPEN:gDPM:gDPMvr, is 1:2:66 in a homogenous water phantom, indicating that variance reduction techniques can significantly increase the calculation efficiency.

Table 1. Performance benchmarks in a homogeneous water phantom.

	gPEN			gDPM			gDPMvr			Relative η		
FS/cm^2	T (min)	σ	η	T (min)	σ	η	T (min)	σ	η	gPEN	gDPM	gDPMvr
2.1 × 2.1	30.01	0.52	0.12	28.44	0.5	0.14	3.39	0.23	5.58	1.00	1.14	45.25
4.2 × 4.2	48.92	0.39	0.13	47.2	0.38	0.15	4.51	0.17	7.67	1.00	1.09	57.09
10.5 × 10.5	299.47	0.34	0.03	87.51	0.33	0.10	14.3	0.16	2.73	1.00	3.63	94.57
Average										1.00	1.96	65.63

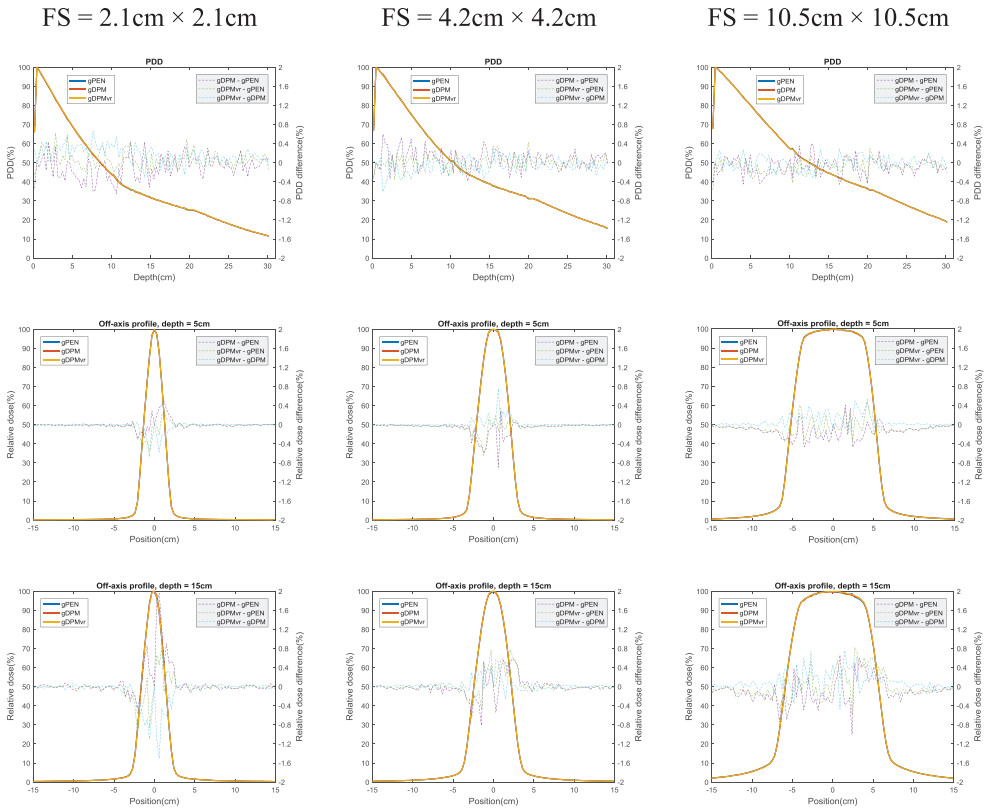


Figure 7. Percentage depth dose (upper row) and off-axis profiles (middle row: 5 cm depth and inside the water, lower row: 15 cm depth and indide the lung) for a water-lung-water phantom. FS: field size.

3.2. Water-lung-water phantom

Figure 7 shows comparisons of PDD and off-axis profiles in a water-lung-water phantom for three different field sizes. The off-axis profiles at 5 cm depth (inside the water) generated from the three algorithms agree with each other to within 0.8%. However, the off-axis profiles at 15 cm depth (inside the lung slab) show a slightly larger difference of around 1%. The PDD profiles become unsmooth at the water-lung interface due to the electron-return-effect (ERE). The z-score histograms (figure 8) exhibit similar patterns as those shown in figure 6, except for slightly more noticeable deviations from being Gaussian with larger σ_{tot} values. The introduction of heterogeneity triggers the ERE, thus augmenting the dose differences between

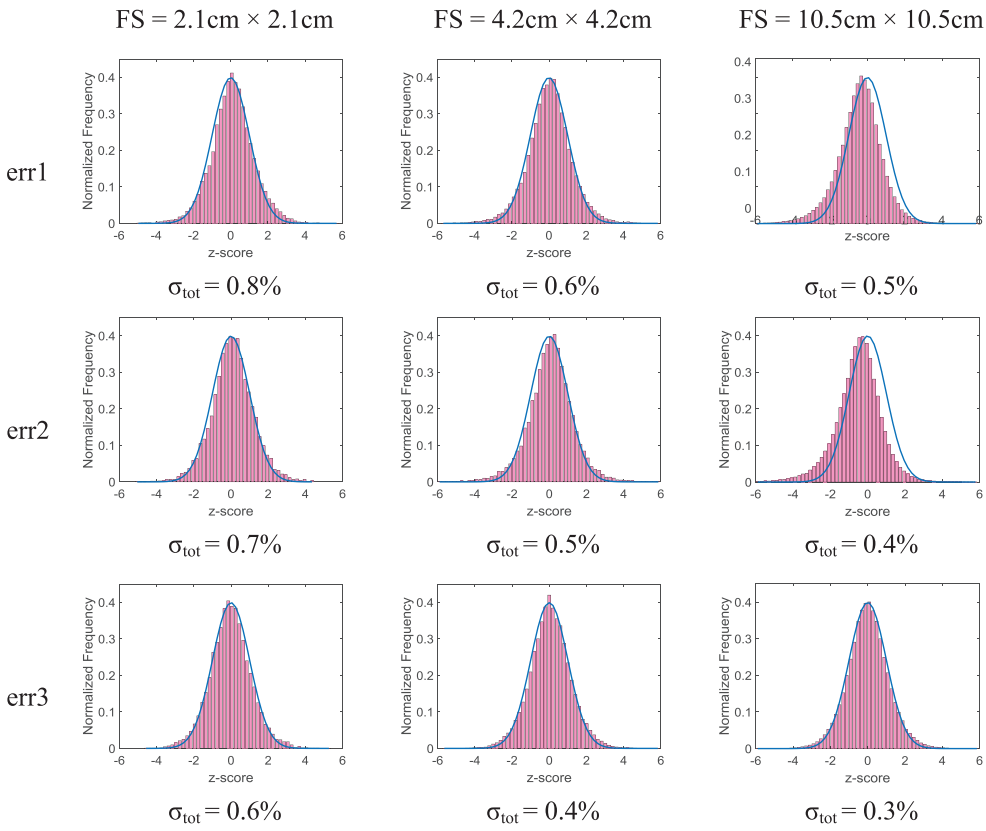


Figure 8. z-score histograms among gPEN, gDPM and gDPMvr for a water-lung-water phantom.

Table 2. Performance benchmarks in a water-lung-water phantom.

	gPEN			gDPM			gDPMvr			Relative η		
FS	T (min)	σ	η	T (min)	σ	η	T (min)	σ	η	gPEN	gDPM	gDPMvr
2.1 cm	27.57	0.59	0.10	36.94	0.55	0.09	3.16	0.26	4.68	1	0.86	44.93
4.2 cm	45.45	0.45	0.11	42.53	0.42	0.13	3.99	0.2	6.277	1	1.23	57.67
10.5 cm	276.42	0.37	0.03	81.26	0.35	0.10	15.9	0.16	2.46	1	3.80	92.97
Average										1	1.96	65.19

gPEN and gDPM/gDPMvr. Nevertheless, the latter two algorithms still show almost identical statistical behaviors.

Table 2 lists the performances achieved by the three algorithms in the water-lung-water phantom. The history number for gPEN and gDPM remains at 10^9 histories while gDPMvr still runs 10^8 histories. The mean efficiency ratio, gPEN:gDPM:gDPMvr, remains almost unchanged (1:2:65) in a more heterogeneous phantom.

3.3. Water-lung-tumor-water phantom

Figure 9 shows profile comparisons in the water-lung-tumor-water phantom. With the introduction of a tumor cube, the PDDs and off-axis profiles at depths of 5 and 12 cm show similar

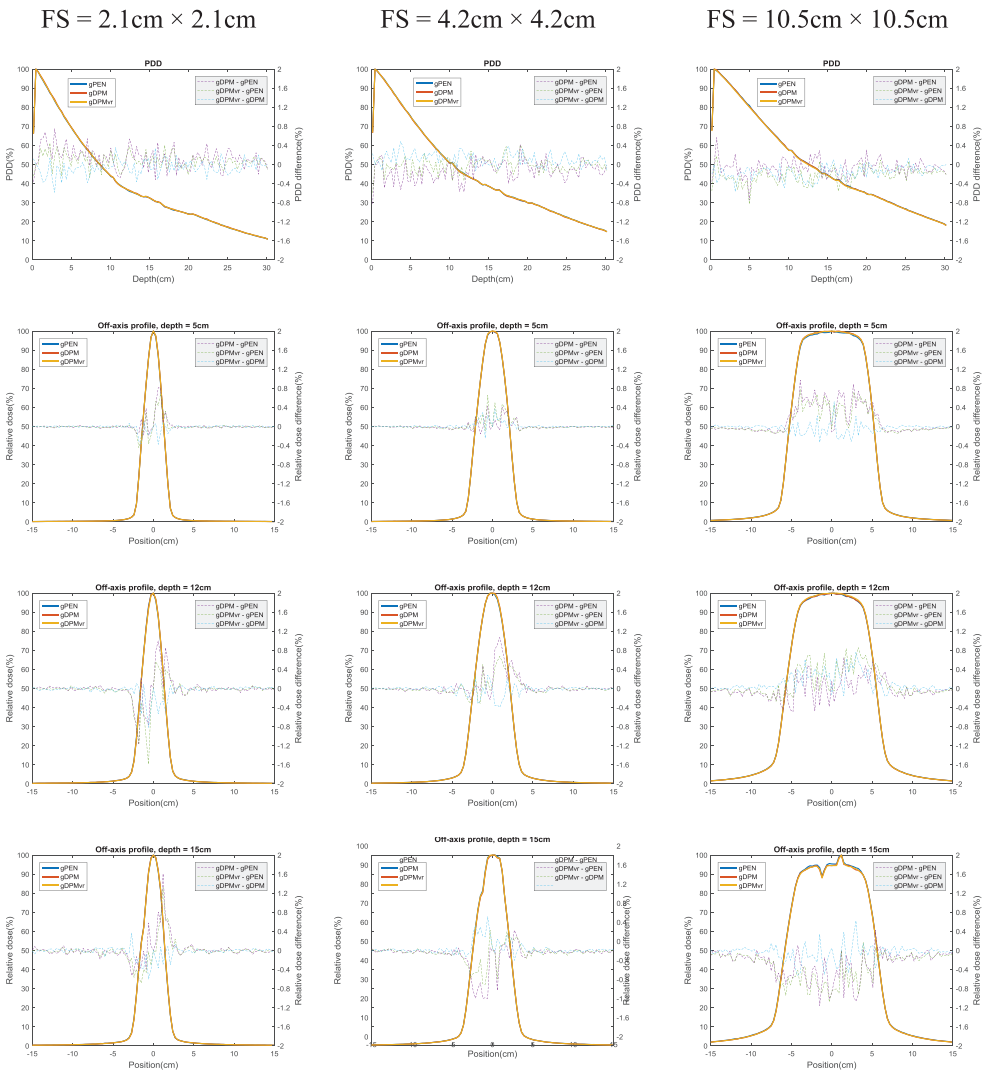


Figure 9. Percentage depth dose and off-axis profile for a water-lung-water phantom. First row: percentage depth dose. Second to fourth rows: off-axis profiles at depth of 5, 12 and 15 cm, i.e. in the water, the lung, the tumor respectively. FS: field size.

patterns as shown in figure 7. However, the off-axis profiles at 15 cm depth traversing both the tumor and the lung show obvious ERE. As shown in figure 10, z-score histogram plots are similar to those shown in figures 6 and 8 except that the observed systematic differences for a field size of 10 cm × 10 cm are less severe, which may be explained by the ERE effect better localizing dose deposition, thus offsetting the adverse effects of ignoring the below-threshold energy photons.

Table 3 summarizes the performances achieved by the three algorithms in the water-lung-tumor-water phantom. The history numbers remain the same as above and a similar efficiency ratio, gPEN:gDPM:gDPMvr, is observed (1:2:58), although gDPMvr's efficiency decreases in the more heterogeneous phantom.

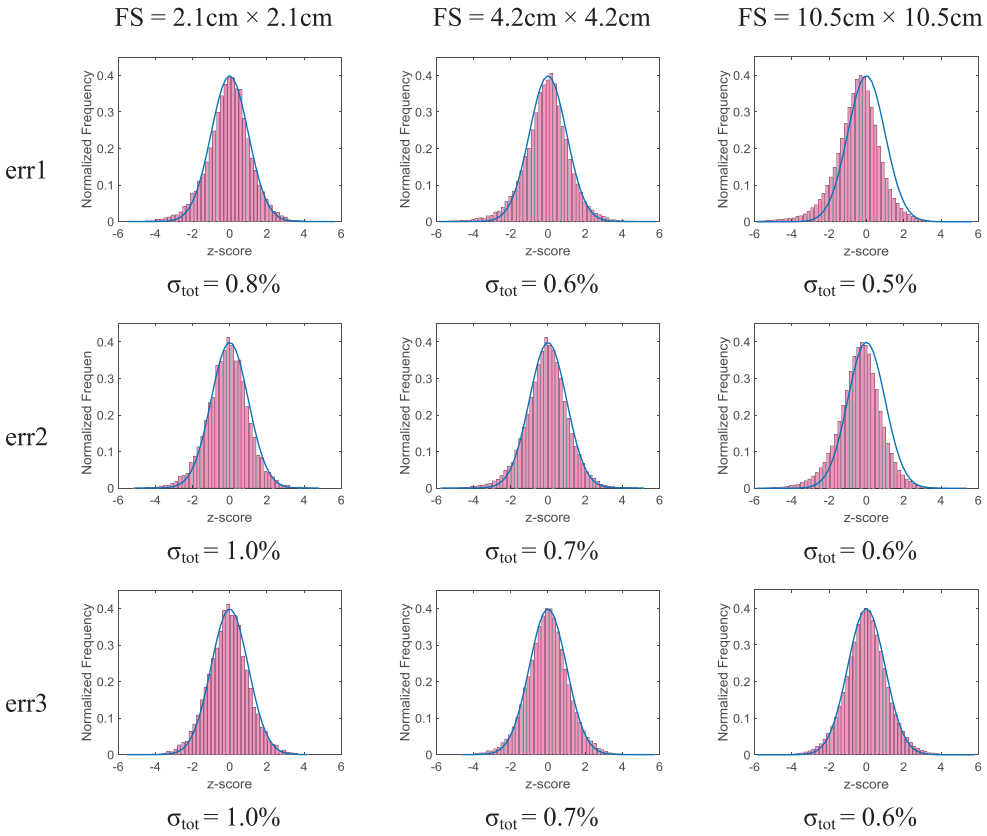


Figure 10. z-score histograms among gPEN, gDPM and gDPMvr for a water-lung-tumor-water phantom.

Table 3. Performance benchmarks in a water-lung-tumor-water phantom.

	gPEN			gDPM			gDPMvr			Relative η		
FS	T (min)	σ	η	T (min)	σ	η	T (min)	σ	η	gPEN	gDPM	gDPMvr
2.1 cm	27.63	0.59	0.10	27.3	0.56	0.12	3.56	0.26	4.16	1	1.12	39.97
4.2 cm	47.09	0.45	0.10	47.46	0.42	0.12	4.43	0.2	5.64	1	1.14	53.81
10.5 cm	281.37	0.37	0.03	85.51	0.35	0.10	18.56	0.16	2.10	1	3.68	81.07
Average										1	1.98	58.28

3.4. Clinical patients

Table 4 shows the gamma passing rates with strict criteria (DTA $= 0$ mm, $\Delta D = 0.5\%D_{\max}$, $D > 10\%D_{\max}$ threshold) and standard deviations of relative differences among gPEN, gDPM and gDPMvr for 15 IMRT treatment plans. The selected z-score histograms

Table 4. Gamma passing rates and standard deviations of relative differences for 15 clinical IMRT plans.

	err1		err2		err3	
Treatment site	γ (%)	σ_{tot} (%)	γ (%)	σ_{tot} (%)	γ (%)	σ_{tot} (%)
Liver	99.90	0.9	99.97	0.8	99.99	0.7
Liver	98.39	0.9	99.21	0.8	99.82	0.7
Liver	99.84	1.0	99.94	0.8	99.98	0.8
Adrenal	99.18	0.9	99.52	0.8	99.92	0.8
Adrenal	99.31	1.0	99.63	0.9	99.90	0.9
Lung	99.70	0.8	99.83	0.7	99.97	0.7
Lung	99.77	0.7	99.88	0.7	99.99	0.6
Mediastinum	99.52	1.0	99.73	0.9	99.97	0.9
Pancreas	99.66	0.7	99.78	0.6	99.99	0.6
Pancreas	97.86	1.1	98.82	0.9	99.63	0.9
Spleen	98.82	0.9	99.49	0.8	99.84	0.7
Stomach	97.76	1.0	98.40	0.9	99.63	0.8
Stomach	97.43	1.0	98.56	0.8	99.61	0.8
Stomach	98.62	0.8	99.36	0.7	99.85	0.7
Stomach	98.24	0.9	98.93	0.8	99.70	0.8

shown in figure 11 reveal systematic differences between gPEN and gDPM/gDPMvr. Nevertheless, the systematic difference is not big with $\overline{\sigma_{\text{tot}}} = 0.9\%$ averaged over all 15 plans. gDPM and gDPMvr, on the other hand, share almost the same statistical behavior. The average gamma passing rate is 98.9% between gPEN and gDPM, 99.4% between gPEN and gDPMvr, and 99.9% between gDPM and gDPMvr. In other words, the chances for dose differences to exceed $0.5\%D_{\text{max}}$ are as small as 1.1% (err1), 0.6% (err2) and 0.1% (err3), respectively.

Table 5 lists the performances achieved by the three algorithms in real patient phantoms. Here we set the termination condition as reaching 1% uncertainty instead of finishing a given history number in order to imitate the real treatment planning system. Moreover, the source particle reuse is toggled on to maximize performance. The mean efficiency ratio of gPEN:gDPM:gDPMvr is about 1:7:43. That is, in highly heterogeneous phantoms, gDPM suffers less performance loss than gPEN due to its much simplified code. The variance reduction scheme can increase calculation efficiency from gDPM by as much as six fold on average, with almost the same accelerating factor being achieved from gPEN to gDPM. gDPMvr can finish calculating a treatment plan in 2.3 min on average with only 0.5% accuracy loss compared to the 'golden standard' gPEN. Thus, gDPMvr is well-suited to fulfill the purpose of verifying adaptive treatment plans in a fast and accurate way.

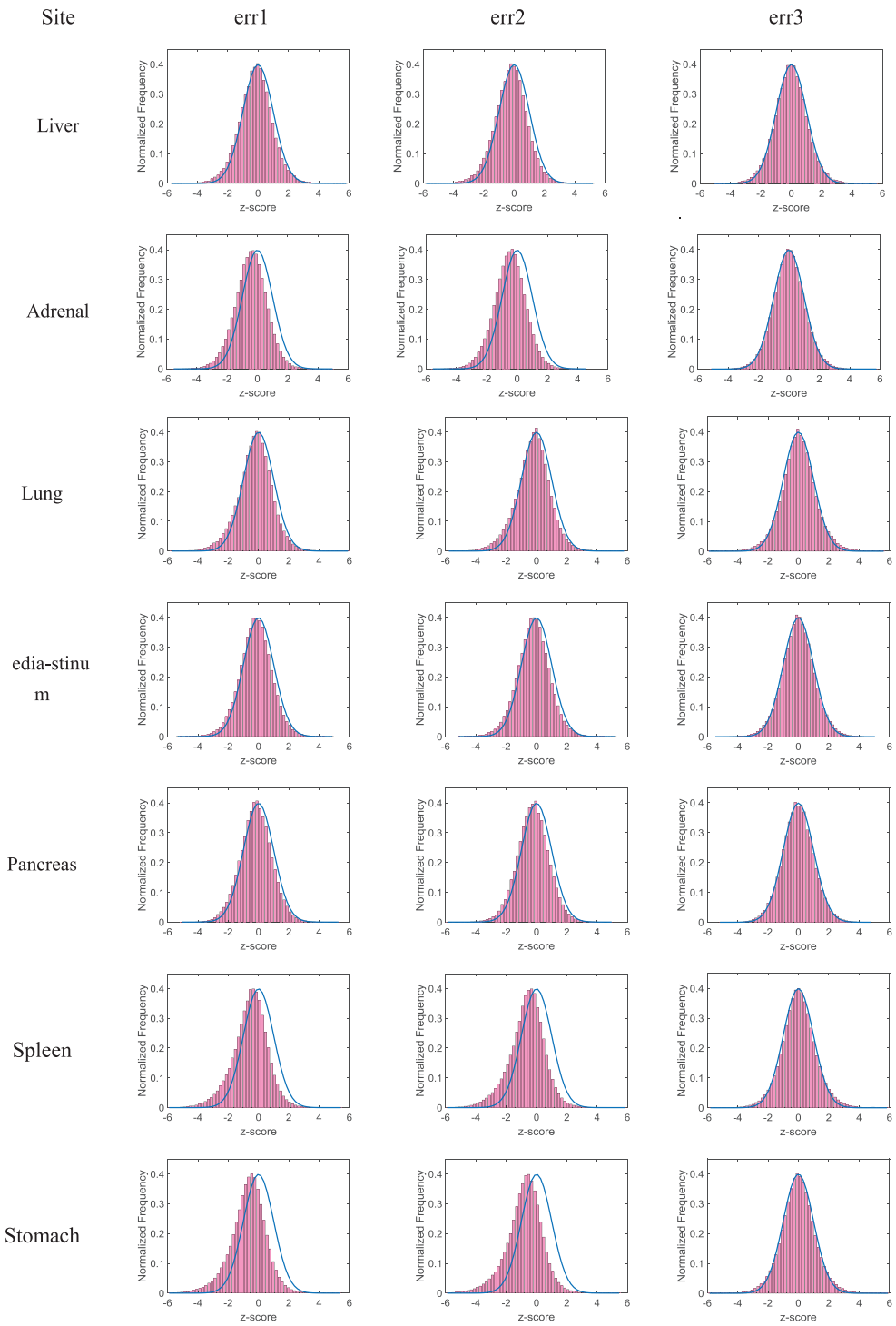


Figure 11. z-score histograms for 7 IMRT plans.

Table 5. Performance benchmarks in real patient phantoms.

	gPEN			gDPM			gDPMvr			Relative η		
FS	T (min)	σ	η	T (min)	σ	η	T (min)	σ	η	gPEN	gDPM	gDPMvr
Liver	41.83	1.12	0.019	9.28	1.08	0.092	1.16	1.07	0.753	1.00	4.85	39.51
Liver	86.85	1.45	0.005	17.28	1.2	0.040	2.52	1.15	0.300	1.00	7.34	54.79
Liver	22.49	1.25	0.028	3.71	1.13	0.211	0.61	1.1	1.355	1.00	7.42	47.61
Adrenal	61.09	1.42	0.008	10	1.2	0.069	2.21	1.13	0.354	1.00	8.55	43.65
Adrenal	77.44	1.1	0.011	24.21	1.04	0.038	3.21	1.03	0.294	1.00	3.58	27.52
Lung	37.98	1.41	0.013	8.79	1.15	0.086	1.56	1.09	0.540	1.00	6.50	40.74
Lung	48.88	1.37	0.011	10.06	1.19	0.070	2.08	1.12	0.383	1.00	6.44	35.16
Mediastinum	37.65	1.37	0.014	7	1.19	0.101	1.59	1.15	0.476	1.00	7.13	33.61
Pancreas	33.65	1.9	0.008	5.32	1.58	0.075	1.36	1.46	0.345	1.00	9.15	41.90
Pancreas	38.5	1.37	0.014	7.03	1.2	0.099	1.03	1.17	0.709	1.00	7.14	51.25
Spleen	78.49	1.46	0.006	14.11	1.21	0.048	2.1	1.18	0.342	1.00	8.10	57.22
Stomach	120.73	1.33	0.005	21.49	1.15	0.035	4.99	1.1	0.166	1.00	7.51	35.37
Stomach	125.55	1.37	0.004	24.67	1.15	0.031	3.41	1.12	0.234	1.00	7.22	55.09
Stomach	101.16	1.33	0.006	22.25	1.14	0.035	2.69	1.11	0.302	1.00	6.19	53.99
Stomach	110.13	1.19	0.006	21.92	1.08	0.039	4.15	1.06	0.214	1.00	6.10	33.45
Average	**68.2**			**13.8**			**2.3**			1.00	6.88	43.39

4. Discussion and conclusion

The recent clinical use of the MRIdian radiation therapy system represents a significant advance in cancer care, enabling clinicians to deliver highly conformal IMRT with real-time MRI guidance. More importantly, the advent of online soft tissue image guidance enables delivery of online adaptive radiation therapy, which is a dramatic departure from conventional treatments employing a single static plan throughout the entire treatment course (Acharya *et al* 2016). In a recent Point/Counterpoint debate (van Vulpen and Wang 2016), it has even been proposed that within the next few years, adaptive hypofractions will become the most common form of radiation therapy. However, this potential paradigm-changing treatment scheme challenges the clinician's ability to assure the safe delivery of the online re-optimized and re-calculated IMRT treatment plans (Noel *et al* 2014), particularly where the dose deposition is subject to a magnetic field. The magnetic field exerts a force on secondary electrons that complicates dose deposition in highly heterogeneous phantoms such as the human body. Monte Carlo is the preferred form of calculation for achieving adequate uncertainty under these challenging conditions. To facilitate the wide adoption of online adaptive radiation therapy that can benefit many patients, development of tools such as rapid and accurate Monte Carlo dose verification is a pressing requirement (Noel *et al* 2014).

Recently we developed a GPU-accelerated Monte Carlo C++ code based on the venerable PENELOPE system, namely gPENELOPE (Wang *et al* 2016). In this work, we accelerated Monte Carlo dose calculation with parallel computation while maintaining original accuracy, with the intention of deploying the platform for complementing experimental dosimetry for treatment subject to a permanent magnetic field which is limited by measurement uncertainty, dimensionality and spatial resolution (Li *et al* 2015). We validated gPENELOPE according to AAPM TG-105 (Chetty *et al* 2007) guidelines by virtue of a number of measurements with both homogenous and heterogeneous phantoms. An acceleration factor of 152 was demonstrated in comparison to the original single-thread FORTRAN implementation with the original accuracy being preserved. Despite this drastic acceleration, the code remains not fast enough for online quality assurance (Acharya *et al* 2016).

Recently Acharya *et al* (2016) reported that the median time for online ART including recontouring, reoptimization, and QA is 26 min for their institution's first patients treated via online ART, with recontouring being the most time-consuming aspect of the procedure. Any Monte Carlo platform for QA thus should require at most several minutes for completion in order to contribute meaningfully to the ultimate goal of minimizing the time required by the ART workflow. DPM, developed by Sempau *et al* (2000), was a major milestone in the development of fast Monte Carlo code for routine clinical use. DPM significantly accelerates Monte Carlo simulation largely by simplifying various charged-particle transport mechanisms. Jia *et al.* later introduced gDPM which accelerated DPM through deployment on a GPU (Jia *et al* 2010). While gDPM substantially accelerates DPM, we found gDPM—adapted to incorporate the MRIdian head model and external magnetic fields—is not adequately fast for implementation in our institution's ART workflow particularly when considering a threshold of 1% *local* uncertainty (versus the 1% *global* average uncertainty used in Jia's work). Achieving 1% local uncertainty is imperative especially in hypofractionated deliveries that include high maximum doses with steep gradients (Acharya *et al* 2016, van Vulpen and Wang 2016). Without adequate reduction in local uncertainty, poor understanding of dose gradients could have severe clinical consequences such as acute toxicities in normal tissues.

In this study, we built upon gDPM by introducing variance reduction and several system-specific simplifications in order to achieve competitive calculation times for implementation in MRgRT ART. These simplifications stem from the mean photon energy in ^{60}Co decay,

the small magnetic field strength of the imaging system, and the 3 mm voxel size utilized for treatment planning in our clinic. The resulting platform—gDPMvr—increases calculation speed of clinical plans by factors of 43 and 6 relative to gPENELOPE and gDPM respectively while preserving adequate (<1%) statistical uncertainty within regions of dosimetric interest. We demonstrated that gDPMvr can achieve 1% mean *local* uncertainty in the $D > 10\%D_{\max}$ region in 2.3 min on average on one Nvidia K80 GPU card for complicated tri-^{60}Co IMRT plans. ViewRay provides its users with a CPU based Monte Carlo secondary dose calculation engine for online ART plan verification QA. For a typical pancreatic IMRT plan, the computation time for 50 million histories is 18 min on a Windows 7 PC with an Intel Core i7 3770 processor (3.4 GHz base frequency, 4 cores and 16 GB RAM). In contrast, the computation time for gDPMvr is 2 min.

This performance is largely enabled by simplifications of electron transport that facilitate GPU implementations of variance reduction techniques that traditionally suffer from thread divergence and limited register number. In fact, we previously attempted to apply a variance reduction technique proposed by Kawrakow and Fippel (2000) in gPENELOPE, but the implementation actually worsened calculation times. Similar results have been reported on another GPU-based Monte Carlo system, namely GPUMCD (Hissoiny *et al* 2011). Simplification of electron transport using CSDA only may compromise accuracy at tissue/air interfaces—such as at the bowel, esophagus, and skin—in a magnetic field. One strategy could be to implement a mixed scheme that can be applied in the regions that are of particular concern. Also, a post-treatment offline recalculation using gPENELOPE can be performed if desired for assessing problematic sub-volumes in the online calculation.

With the imminent introduction of MRI-guided linac devices (van Vulpen and Wang 2016), integrating variance reduction techniques on GPUs will be a challenging and pressing problem. Moreover, the required accuracy of an online Monte Carlo system and the appropriate QA metrics, e.g. conventional gamma criteria versus dose-volume-histogram metrics (Zhen *et al* 2011), must be established. Nonetheless, gDPMvr achieves speeds beyond those required by in-use workflows for MRgRT ART while preserving accuracy comparable to that achieved by more traditional Monte Carlo code.

Acknowledgments

The authors graciously acknowledge Drs Iwan Kawrakow and James Dempsey of ViewRay, Inc. for many discussions on this subject. The authors declared no potential conflict of interest with respect to the research, authorship, and/or publication of this paper.

References

Acharya S *et al* 2016 Online magnetic resonance image guided adaptive radiation therapy: first clinical applications *Int. J. Radiat. Oncol. Biol. Phys.* **94** 394–403

Ahmad S B, Sarfehnia A, Paudel M R, Kim A, Hissoiny S, Sahgal A and Keller B 2016 Evaluation of a commercial MRI Linac based Monte Carlo dose calculation algorithm with GEANT4 *Med. Phys.* **43** 894–907

Chetty I J *et al* 2007 Report of the AAPM task group no. 105: issues associated with clinical implementation of Monte Carlo-based photon and electron external beam treatment planning *Med. Phys.* **34** 4818–53

Faddegon B A, Kawrakow I, Kubyshin Y, Perl J, Sempau J and Urban L 2009 The accuracy of EGSnrc, Geant4 and PENELOPE Monte Carlo systems for the simulation of electron scatter in external beam radiotherapy *Phys. Med. Biol.* **54** 6151–63

Hissoiny S, Ozell B, Bouchard H and Despres P 2011 GPUMCD: a new GPU-oriented Monte Carlo dose calculation platform *Med. Phys.* **38** 754–64
Jahnke L, Fleckenstein J, Wenz F and Hesser J 2012 GMC: a GPU implementation of a Monte Carlo dose calculation based on Geant4 *Phys. Med. Biol.* **57** 1217–29
Jia X, Gu X, Graves Y J, Folkerts M and Jiang S B 2011 GPU-based fast Monte Carlo simulation for radiotherapy dose calculation *Phys. Med. Biol.* **56** 7017–31
Jia X, Gu X, Sempau J, Choi D, Majumdar A and Jiang S B 2010 Development of a GPU-based Monte Carlo dose calculation code for coupled electron–photon transport *Phys. Med. Biol.* **55** 3077–86
Kawrakow I and Fippel M 2000 Investigation of variance reduction techniques for Monte Carlo photon dose calculation using XVMC *Phys. Med. Biol.* **45** 2163–83
Lagendijk J J, Raaymakers B W, Van den Berg C A, Moerland M A, Philippens M E and van Vulpen M 2014 MR guidance in radiotherapy *Phys. Med. Biol.* **59** R349–69
Li H, Rodriguez V L, Green O L, Hu Y, Kashani R, Wooten H O, Yang D and Mutic S 2015 Patient-specific quality assurance for the delivery of co intensity modulated radiation therapy subject to a 0.35-T lateral magnetic field *Int. J. Radiat. Oncol. Biol. Phys.* **1** 65–72
Low D A, Harms W B, Mutic S and Purdy J A 1998 A technique for the quantitative evaluation of dose distributions *Med. Phys.* **25** 656–61
Malkov V N and Rogers D W 2016 Charged particle transport in magnetic fields in EGSnrc *Med. Phys.* **43** 4447–58
Noel C E, Santanam L, Parikh P J and Mutic S 2014 Process-based quality management for clinical implementation of adaptive radiotherapy *Med. Phys.* **41** 081717
Raaijmakers A J, Raaymakers B W and Lagendijk J J 2008 Magnetic-field-induced dose effects in MR-guided radiotherapy systems: dependence on the magnetic field strength *Phys. Med. Biol.* **53** 909–23
Rankine L J, Mein S, Cai B, Curcuru A, Juang T, Miles D, Mutic S, Wang Y, Oldham M and Li H 2017 Three-dimensional dosimetric validation of a magnetic resonance-guided intensity modulated radiation therapy system *Int. J. Radiat. Oncol. Biol. Phys.* **97** 1095–104
Sempau J, Wilderman S J and Bielajew A F 2000 DPM, a fast, accurate Monte Carlo code optimized for photon and electron radiotherapy treatment planning dose calculations *Phys. Med. Biol.* **45** 2263–91
van Vulpen M and Wang L 2016 Within the next five years, adaptive hypofractionation will become the most common form of radiotherapy *Med. Phys.* **43** 3941
Wang Y *et al* 2016 A GPU-accelerated Monte Carlo dose calculation platform and its application toward validating an MRI-guided radiation therapy beam model *Med. Phys.* **43** 4040–52
Woodcock E, Murphy T, Hemmings P and Longworth S 1965 Techniques used in the GEM code for Monte Carlo neutronics calculation *Proc. Conf. Applications of Computing Methods to Reactors ANL-7050*
Zhen H, Nelms B E and Tome W A 2011 Moving from gamma passing rates to patient DVH-based QA metrics in pretreatment dose QA *Med. Phys.* **38** 5477–89