Received: 28 January 2021 | Revised: 20 September 2021 | Accepted: 6 October 2021

DOI: 10.1002/mp.15291

RESEARCH ARTICLE

MEDICAL PHYSICS

XIORT-MC: A real-time MC-based dose computation tool for low- energy X-rays intraoperative radiation therapy

Paula Ibáñez[1,2] | **Amaia Villa-Abaunza**[1] | **Marie Vidal**[1,3] | **Pedro Guerra**[4,5,6] | **Sergio Graullera**[7] | **Carlos Illana**[7] | **José Manuel Udías**[1,2]

[1] Nuclear Physics Group, EMFTEL and IPARCOS, CEI Moncloa, Universidad Complutense de Madrid, Madrid, Spain

[2] Instituto de Investigación Sanitaria del Hospital Clínico San Carlos, Madrid, Spain

[3] Department of Radiotherapy, Centre Antoine-Lacassagne, Nice, France

[4] Department of Electronic Engineering, ETSIT, CEI Moncloa, Universidad Politécnica de Madrid, Madrid, Spain

[5] Biomedical Research Center in Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN), Madrid, Spain

[6] Tres Cantos, MedLumics S.L., Madrid, Spain

[7] Tres Cantos, GMV, Madrid, Spain

Correspondence
Paula Ibáñez, Nuclear Physics Group, EMFTEL and IPARCOS, CEI Moncloa, Universidad Complutense de Madrid, 28040 Madrid, Spain.
Email: pbibanez@ucm.es

Funding information
Spanish Government, Grant/Award Numbers: RTI2018-098868-B-I00, RTC-2019-007112-1, XPHASE-LASER; Comunidad de Madrid, Grant/Award Numbers: B2017/BMD-3888 PRONTO-CM, IND2018/BMD-9990

Abstract

Purpose: The INTRABEAM system is a miniature accelerator for low-energy X-ray Intra-Operative Radiation Therapy (IORT), and it could benefit from a fast and accurate dose computation tool. With regards to accuracy, dose computed with Monte Carlo (MC) simulations are the gold standard, however, they require a large computational effort and consequently they are not suitable for real-time dose planning. This work presents a comparison of the implementation on Graphics Processing Unit (GPU) of two different dose calculation algorithms based on MC phase-space (PHSP) information to compute dose distributions for the INTRABEAM device within seconds and with the accuracy of realistic MC simulations.

Methods: The MC-based algorithms we present incorporate photoelectric, Compton and Rayleigh effects for the interaction of low-energy X-rays. XIORT-MC (X-ray Intra-Operative Radiation Therapy Monte Carlo) includes two dose calculation algorithms; a Woodcock-based MC algorithm (WC-MC) and a Hybrid MC algorithm (HMC), and it is implemented in CPU and in GPU. Detailed MC simulations have been generated to validate our tool in homogeneous and heterogeneous conditions with all INTRABEAM applicators, including three clinically realistic CT-based simulations. A performance study has been done to determine the acceleration reached with the code, in both CPU and GPU implementations.

Results: Dose distributions were obtained with the HMC and the WC-MC and compared to standard reference MC simulations with more than 95% voxels fulfilling a 7%-0.5 mm gamma evaluation in all the cases considered. The CPU-HMC is 100 times more efficient than the reference MC, and the CPU-WC-MC is about 50 times more efficient. With the GPU implementation, the particle tracking of the WC-MC is faster than the HMC, with the extraction of the particle's information from the PHSP file taking a major part of the time. However, thanks to the variance reduction techniques implemented in the HMC, up to 400 times less particles are needed in the HMC to reach the same level of noise than the WC-MC. Therefore, in our implementation for INTRABEAM energies, the HMC is about 1.3 times more efficient than the WC-MC in an NVIDIA GeForce GTX 1080 Ti card and about 5.5 times more efficient in an NVIDIA GeForce RTX 3090. Dose with noise below 5% has been obtained in realistic situations in less than 5 s with the WC-MC and in less than 0.5 s with the HMC.

Conclusions: The XIORT-MC is a dose computation tool designed to take full advantage of modern GPUs, making possible to obtain MC-grade accurate dose distributions within seconds. Its high speed allows a real-time dose calculation that includes the realistic effects of the beam in voxelized geometries

of patients. It can be used as a dose-planning tool in the operating room during a XIORT treatment with any INTRABEAM device.

KEYWORDS
dose calculation, GPU, intraoperative radiation therapy, Monte Carlo

1 | INTRODUCTION

The use of Intra-Operative Radiation Therapy (IORT) with low-energy X-rays is increasing rapidly in the clinical field. The INTRABEAM system (Carl Zeiss Surgical Gmbh, Oberkochen, Germany) is a miniature low-kV X-ray generator developed for IORT. Electrons are accelerated to energies of 50 keV through a narrow probe and hit a gold target, generating X-rays in a quasi-isotropic way.[1–3] Different applicators may be attached to the probe. Spherical applicators are the most used, mainly for partial breast irradiation[4,5] but also for treating other locations such as glioblastomas.[6] A needle applicator is used for treating spinal metastasis or brain tumors,[7,8] and flat and surface applicators are used mainly for superficial treatments.[9]

However, this technique is currently limited by the inherent difficulties associated with the planning process, as the treatment plan is usually done in situ just after the tumor resection inside the operating room. Therefore, planning is normally based on isodose curves in water, which may lead to large inaccuracies when heterogeneous media are treated.[10–12]

Among all the algorithms developed for dose calculation, Monte Carlo (MC) is considered the gold standard. However, if high accuracy is sought, it is required simulating many particles and consequently large computation times, which is not compatible with real-time dose estimations as needed for IORT. Fast MC codes have been developed for dose calculations, such as DPM,[13] which includes a precalculation stage to increase the efficiency by 50 times when compared to a regular MC.[14] Other codes are parallelized and perform the computation in CPU clusters. Tyagi et al. implemented DPM to work in parallel in a cluster,[15] while Ziegenhein et al. developed PhiMC, which is another implementation of DPM for parallel execution within one shared memory node in a multicore CPU.[16] The main disadvantage of these codes is the requirement of a high-capacity cluster to perform the calculations.

An alternative approach is to use graphic processing units (GPUs). GPUs are easy to access and maintain, as they can run on a local computer, and they are much less expensive than a CPU cluster. A GPU can execute simultaneously hundreds of threads of a program, and consequently, algorithms can be drastically accelerated. There have been several works to accelerate MC codes using GPU.[17] Badal and Badano developed a MC in GPU (MC-GPU) for photons in voxelized geometries[18] obtaining a maximum 27-fold speed-up factor when compared to PENELOPE.[19,20] Jia et al. developed a GPU version of DPM[21] reaching speed-up factors of about five to six times for megavoltage photons and electrons. Hissoiny et al. developed the GPUMCD,[14] a GPU-oriented MC dose calculation platform suitable for radiotherapy treatments two orders of magnitude faster than DPM. Yu et al. have developed an OpenCL-based MC code for photon transport (MCX-CL),[22] achieving a 10% average performance improvement on NVIDIA GPUs, and Su et al. developed the ARCHER$_{RT}$ code[23] for tomotherapy, reaching calculation rates of about 10 million particles per second.

However, the INTRABEAM accelerator delivers dose distributions with very high gradients. This makes the dose calculation very inefficient, especially in cases where shielding or low-density tissues are involved. In these cases, the number of histories required to reach reasonably small statistical noise levels in these areas will be very high, increasing the simulation time, even if a GPU-MC is involved.

In previous works, we very briefly introduced a new algorithm running in CPU, named Hybrid MC (HMC),[24–28] to calculate dose deposited by the INTRABEAM device within minutes, fully taking into account the different tissues and structures of the patient, that is, as derived from a 3D Computed Tomography (CT) image. The HMC includes all the relevant physics of a regular MC minimizing the contribution of statistical noise, so dose distributions may be obtained in a CPU in a much shorter time than a regular full-MC simulation. Thanks to variance reduction techniques incorporated in the code, the number of histories needed to run a HMC calculation is much smaller than the one needed in a regular MC, for the same statistical noise. Although some results of this algorithm have been presented before, its inner workings have never been described in detail. The optimizations in the HMC algorithm are strongly oriented toward efficient execution in a CPU, with a very dense access to memory and reduced number of random number generations calls. Thus, the question arises about whether some of the strategies introduced in the HMC, such as continuous forced interactions, may not be well suited for execution in GPUs and whether other algorithms might be more suitable when targeting GPU execution.

In this work, the HMC is described in detail for the first time. The general features of the code and the novel variance reduction techniques are fully detailed. Moreover, we have further accelerated the HMC and we

have implemented a HMC package on GPU architecture using CUDA Fortran. The implementation in GPU is also described here in detail. To assess the efficiency of the HMC algorithm and its suitability to GPUs, another GPU-based MC algorithm based on the Woodcock particle tracking,[29] the Woodcock MC (WC-MC), has also been ported to GPU with the same tools. Both algorithms have been embedded in a tool to calculate dose distributions for INTRABEAM. It can be executed in CPU or GPU allowing to choose between the MC-based algorithm (WC-MC) and the hybrid algorithm (HMC). The implementation of both codes in GPU makes real-time dose calculations possible. A dosimetric evaluation against reference MC simulations has been performed to prove the accuracy of both the HMC and the WC-MC. Moreover, a detailed efficiency study against CPU- and GPU-based codes has been performed to establish the acceleration reached. To assess the effect of the evolution of GPU in the efficiency of the implementations, two different GPUs architecture (Ampere and Pascal) from NVIDIA have been benchmarked, one state-of-the-art and another one from 4 years ago.

2 | MATERIAL AND METHODS

2.1 | General features

The XIORT-MC (X-ray Intra-Operative Radiation Therapy MC) is a MC-based tool for the transport of INTRABEAM X-rays across voxelized geometries. Physics from PENELOPE[20] has been incorporated, as PENELOPE models are very accurate for kilo-voltage X-rays.[30–32] Tables with the attenuation coefficients for the different materials have been precalculated based on PENELOPE to speed up calculations.

The user can select the dose computation algorithm and the CPU or GPU execution. Two codes have been incorporated in XIORT-MC, the HMC and the WC-MC. Both codes are described in detail in the following sections.

2.2 | Woodcock MC (WC-MC)

The WC-MC algorithm was included in the XIORT-MC as a reference (or more conventional) MC-based algorithm. The algorithms and cross-sections used in our implementation are very similar to the ones in MC-GPU,[18] as both of them are PENELOPE-based. The main difference of the current implementation was the use of the precalculated files with energies and angles for the Compton interaction and angular tables for the Rayleigh scattering.

The WC-MC is a MC algorithm where particles travel through a voxelized volume with the Woodcock tracking algorithm.[29] This algorithm assumes that the whole volume is homogeneously filled with the most attenuating material present in the phantom. This overattenuation is compensated by the introduction of "virtual (no-)interactions." This way, instead of reading the voxel composition each time the particle crosses a voxel, the distance to the next interaction is randomly sampled, and the code determines voxel material and density only after the particle is transported.

At each history update, the photon can undergo photoelectric, Compton, Rayleigh, or virtual interactions. If any of the first three interactions take place, the particle is removed from the primary beam and, in the case of Compton and Rayleigh interactions, secondary particles are created: a secondary photon in the case of Rayleigh interaction and a scattered secondary photon and one electron in the case of Compton. If, on the other hand, a virtual interaction occurs, the particle continues traveling through the volume with the same energy and trajectory. In view of the relatively small energies involved in the INTRABEAM, secondary electrons generated in Compton and photoelectric events are simulated with zero range.

2.3 | Hybrid Monte Carlo (HMC)

The HMC is a dose computation algorithm with several variance reduction techniques to speed up calculations. A detailed description of these techniques is presented in this section.

2.4 | Meta-histories approximation

The meta-histories (m-histories) approximation[24,28] was used to track the particles throughout the volume. Photons were condensed and studied macroscopically, which presented less computational cost than the conventional approach of regular MC codes that consisted of studying each photon interaction separately. In the macroscopic approach, each m-history represented an arbitrary number of photons and carried a weight proportional to the number of photons that should be generated at the source with the given characteristics (energy, position, emission angle). The accumulated probability of interaction was scored after moving a given step size. After each step, the new weights for the m-history were computed, taking into account the weight lost at the previous interactions. The dose was computed from the accumulated reduction of the weight of the m-histories as they traveled through the region.

Thus, condensed histories (m-histories) for photons at the source were generated from the information stored in phase-space (PHSP) files and transported with a fixed energy throughout the volume in steps of length *dl*, whose value was typically smaller than half the voxel size. At each step, a fraction of the m-history

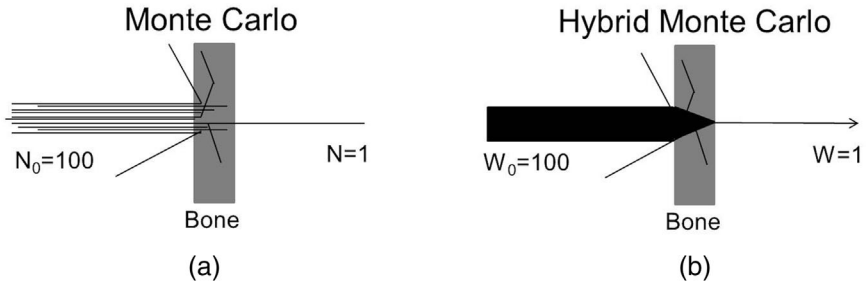

FIGURE 1 (a) Schematic representation of 100 photons travelling through a bone layer in a regular MC and (b) the equivalent representation in the HMC of one meta-history with a weight of 100 photons. Note that in the regular MC only one photon survives after attenuation in the bone layer while in the HMC the same number of primary m-history exits the bone, but with a weight that is 1 of 100 of the initial weight

interacted, being absorbed by photoelectric effect or being scattered by Compton or Rayleigh effects. The weight of each condensed history was updated as it traveled through matter. This way, the number of m-histories would remain constant along the volume, being only its weight reduced at each interaction, so that the number of m-histories would not decrease far from the source due to photon absorption. For instance, beam hardening would be reflected in a reduction of the weights of low-energy particles compared to the ones of higher energy, but the number of m-histories of each energy would be kept constant throughout the simulation. Smaller energy particles would not be absorbed inside the volume, but their weight would be decreased up to the point that it becomes vanishingly small, so they would not contribute to the final dose.

Therefore, problems associated with low statistics (low number of histories) would be reduced, as we had always the same number of primary virtual histories, regardless of how much attenuation the material offered. Secondary particles would be generated inheriting the weights of the primary radiation of origin, and all the relevant phenomena related to interaction of radiation in media, such as backscatter or build-up, would be accurately reproduced. A schematic representation of the difference in particle tracking in a regular MC and in the HMC is shown in Figure 1.

2.5 | Interaction forcing

The m-history would always interact via photoelectric, Compton and Rayleigh interactions after each history update. The loss of weight (average flux loss) after every interaction of the m-history was computed in a continuous way as $1\text{-}e^{-\mu dl}$, where dl is the spatial displacement in the direction of the m-history since the last update and μ is the attenuation coefficient for photoelectric, Rayleigh or Compton effects, taken from the tables of PENELOPE,[20] at the spatial point where the updates of m-history position were being done.

The weight associated to the scattered or attenuated photons was subtracted from the weight of the primary m-history, and secondary histories were produced with their corresponding weight and energy. The primary m-history kept travelling through the volume in the same direction, but with an updated weight and the same initial energy.

2.6 | Secondary particles

Secondary photons created during Compton and Rayleigh interactions inherited the weight of the primary radiation. Once created, they were no longer treated as m-history, but as regular MC histories. They were emitted in different angles and with different energies, according to the energy-angle relationship for the Compton effect on bound electrons and the angle emission for the Rayleigh scattering,[20] according to the physics of PENELOPE.

In the case of the INTRABEAM, secondary electrons would be emitted with very low energies presenting very low ranges, so local absorption of the electrons was considered. An MC study was performed to evaluate the range of electrons with energies up to 50 keV in different biological materials. For all the energies, more than 99% of electrons were absorbed before 0.25 mm in the different media. Therefore, it was reasonable to consider local absorption of the electrons, provided voxel sizes were not much smaller than 0.25 mm.

2.7 | Dose normalization

A fluence normalization of the primary particles was incorporated to avoid aliasing and undersampling artifacts in the dose image, which may appear if a very low number of m-histories was used.

m-histories were never absorbed inside the region considered, so it was possible to know beforehand the number of m-histories that had travelled through the volume and the number of interactions that had taken place in each voxel. Note that the interactions in each voxel would be independent of the materials the region was made of. Different materials, thus, different absorptions, would only affect the weight, but not the fluence of the number of m-histories. Therefore, for a given source, the number of m-histories in a region would be the same if the region was made of water or of lead, as the material of the voxel would only be reflected on the deposited energy, deriving from the weights, but not on the

fluence. Thus, we could calculate a priori the number of m-histories and interactions that should correspond to each voxel from a given fluence at the source and apply a normalization to the dose to generate an accurate, artifact-free dose using a relative small number of primary m-histories.[24]

The number of interactions inside each voxel was compared with the estimated value based on the known fluence. A normalization procedure involving the actual fluence in the calculation (i.e., how many rays travel through a given voxel for a particular source, including effects of anisotropy and homogeneity) was incorporated in the HMC calculation. The actual number of interactions in each voxel was removed from the dose and replaced by the ideal number of interactions for the given source and a homogeneous material.

That is, the normalized dose in a voxel *i* of any given applicator would be computed as:

$$\text{normalized dose}(i) = \frac{\text{dose}\,(i)}{\text{norm}\,(i)}\text{ideal}(i). \tag{1}$$

where *dose(i)* would be the original dose in the voxel, *norm(i)* would be the normalization value with the actual number of interactions in the voxel, generated from the actual fluence and *ideal(i)* would be the expected number of interactions in the voxel *i* of an identical source travelling through a homogeneous, nonattenuating medium.

In spherical applicators, particles are emitted by a localized source with fluence $f(\theta,\varphi)$ per surface unit. It can be assumed that fluence does not depend on the azimuthal angle φ, that is, that the spherical sources exhibit, at least, one-axis rotational symmetry. This assumption can be removed, if needed, but we will keep it here for the purpose of simplicity in the explanation of the normalization procedure.

The number of primary m-histories passing through a differential area dS is $f(\theta)dS$, being $dS = r^2 sin(\theta)d\theta d\varphi$. If the source is considered isotropic, there is no dependence on θ. This way, the expected fluence can be obtained from a simple analytical expression.

Let us consider a spherical shell covering all the angles. If N rays were generated, the number of rays per surface unit would be $N/4\pi R^2$. On the other hand, if every ray was sampled longitudinally every particle step length dl, there would be dR/dl samples per each ray traversing a spherical shell with radius dR. Therefore, in total there would be $N(dR/dl)$ samples within the spherical shell with radius dR. The density of sampling points within the spherical shell would be:

$$\text{density} = \frac{\text{Number samples shell}}{\text{Shell volume}} = \frac{N\frac{\text{dR}}{\text{dl}}}{4ßR^2\text{dR}} = \frac{N}{4ßR^2\text{dl}}. \tag{2}$$

Therefore, the number of samples corresponding to a voxel *i* with volume f^3, with *f* being the voxel size, would be the density multiplied by the voxel volume, or:

$$\text{Number of samples(i)} = \frac{N \cdot f^3}{4ßR^2\text{dl}} \tag{3}$$

The dose per m-history would be:

$$\left[\frac{\text{dose(i)}}{\text{norm(i)}}\right]\left[\frac{1}{\text{N}}\right]\left[\frac{Nf^3}{4ßR^2\text{dl}}\right]. \tag{4}$$

Finally, the normalized dose could be written as:

$$\text{normalized dose (i)} = \frac{\text{dose(i)}}{\text{norm(i)}}\frac{f^3}{4ßR^2\text{dl}}. \tag{5}$$

In the case of flat and surface applicators, the source is located at a known position inside the applicator and the beam goes through different filters before reaching the surface of the applicator. Therefore, in this case, there were no simple analytical expressions to obtain the expected fluence and the fluence factor was extracted from the PHSP file.

In this work, the effect of the normalization was studied and an evaluation of the statistical noise of normalized and not normalized dose distributions was made to assess the effect of the normalization in the dose.

2.8 | GPU implementation

Both XIORT-MC algorithms were ported to GPU. Our aim was to translate the code to GPU with few deviations from the CPU code as possible to assess what is the benefit and potential of the algorithms from GPU execution. Thus, no GPU-specific optimization was attempted, other than taking care of using the least possible updates of global memory, and to adjust the number of threads and blocks best suited for each model of GPU and algorithm.

2.9 | Technical details and memory management

The XIORT-MC algorithms were implemented in GPU using CUDA Fortran developed by NVIDIA together with The Portland Group. The performance of the simulations on two different GPU cards was studied; an NVIDIA GeForce GTX 1080 Ti card with 3584 cores and 11172 Mbytes of global memory (GPU1) and a NVIDIA GeForce RTX 3090 with 10496 cores and 24260 Mbytes of global memory (GPU2). The HMC and the WC-MC codes were written in standard Fortran. CUDA Fortran extensions were called to initialize the GPU and to move

data inside and outside the GPU memory for the GPU code.

In principle, both the HMC and the WC-MC algorithms were easily amenable to execute in GPU because each primary history is independent of the others. Thus, each primary history and its corresponding secondary particles were calculated in independent threads, so it was possible to simulate hundreds of histories at the same time. However, there are some limitations in the performance of the algorithms that might slow down the calculations. The main limitation, especially in the case of the HMC, was the number of memory writing operations that had to be done in each thread. To ensure a correct performance of the algorithm and to guarantee that two or more threads were not writing information simultaneously, atomic operations had to be introduced so that only one thread was updating the energy image and the normalization volume at a given moment. These atomic operations incurred in a performance penalty. Other possible complications were related to the memory size, because large PHSP files would be needed to obtain very high statistics. To solve this, a parameterization of the PHSP files was employed, as it is explained in the following subsection.

Different memory levels were used when writing the code. The object geometry and some of the data, such as attenuation coefficient information or Rayleigh and Compton scattering angles and energies for 24 different materials (about 200 MB), were stored in texture memory. Each time an m-history was updated, the information of the geometry and material was fetched from the textures. Information regarding PHSP file, deposited energy image or normalization volume was stored in the global device memory. Common parameters to all threads were stored in constant memory, and internal variables to each thread were stored in local memory.

Dose computation was then split among thousands of threads, being each thread in charge of decoding the PHSP, transporting the primary and secondary particles and computing the deposited energy and the normalization volume (in the case of the HMC). Different seeds for the random number generators were provided to each thread during the execution of both codes.

A flowchart with the parallelization process in GPU and the meta-histories transport of the HMC code is shown in Figure 2.

2.10 | Debinning

The XIORT-MC algorithms read the particles information from an external PHSP file precomputed from detailed simulations of the X-ray source and applicators. In order to achieve dose distributions with acceptable level of statistical noise, a very high number of initial particles were required, especially when the WC-MC was chosen. For instance, for the largest spherical applicator, more than $2{\cdot}10^9$ histories were simulated, which would mean PHSP files with sizes above 250 Gb. Even though the memory available in modern GPUs is increasing, it is still limited compared to CPUs and to the one required to store the whole PHSP. Further, even storing several of these huge files on disk would be relatively difficult. Thus, PHSP files were precomputed and parameterized to arrange them into bins removing redundant degrees of freedoms, taking advantage of the rotational symmetry around the longitudinal axis of all the applicators. This also allowed to improve the efficiency of the previous realistic MC simulations of the applicators needed to obtain the PHSP. As a consequence, a debinning process inside XIORT-MC was introduced during dose computation to recover the information required to generate every particle.

In the case of needle and spherical applicators, we followed the parameterization described in Ref. 28, where PHSP files are fully defined by the distribution of particles in energy E and two angles α and β, the first to position the particle on the sphere surface, and the second to determine the direction of emission of the particle. For the case of flat and surface applicators, the parameterized PHSP files were defined by the distribution of particles in energy E, radial distance to the applicator axis ρ, and angle θ, representing the deviation of the particle with respect to the primary angle of the trajectory.[33] The variables employed are shown in Figure 3.

A trade-off between accuracy and number of bins was made to select the optimum bin size. Starting from a reference PHSP with 1000 million particles, it was binned into successively coarser bins in the variables previously described. The dose computed with this PHSP, once debinned, was compared to the one from the original PHSP. With the bin sizes employed in this work, the doses from binned and original PHSP agree well within the 7%-0.5 mm gamma criteria[34] (more than 99.5% of the voxels passed the test). The spherical PHSP files were histogrammed with 50 bins in energy, ranging from 0 to 50 keV, 200 bins in α, from 0° to 180°, and 200 bins in β, ranging from 0° to 20°. The flat and surface PHSP files contained 50 bins in energy, ranging from 0 to 50 keV, 180 bins in the deviation angle θ, from 0° to 18°, and 70 bins in the radial position ρ, ranging from 0 to 3.5 cm.

The XIORT-MC needed to reconstruct the information to generate particles from these compact PHSP files. In the case of spherical applicators, the information from the PHSP file was combined with two randomly chosen azimuthal angles uniformly picked in between 0° and 360°. One of them, combined with α, determined the location of the emission point for the particle in the surface of the sphere. The second one, combined with β, determined the direction of emission of the particle. For flat and surface applicators, a similar approach was employed. Again, two uniformly distributed random angles were generated between 0° and 360°. One of them, defined the position of the particle *(x,y,z)*, in

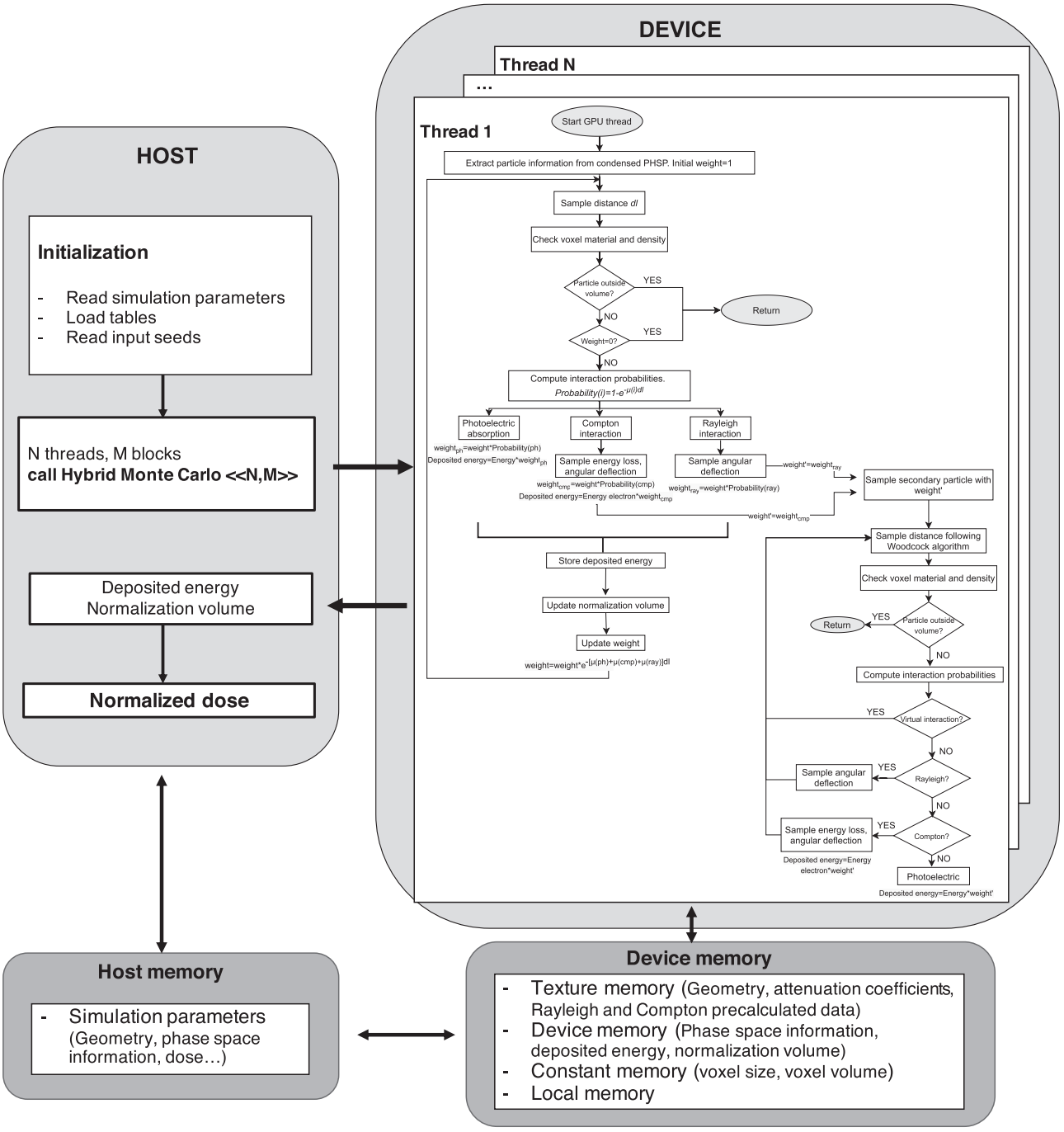

FIGURE 2 Flowchart of the parallel simulation in GPU with the HMC algorithm

combination with the distance to the applicator axis ρ. The other one, combined with the deviation angle θ, defined the direction of the particle. The compact PHSP also contained the relative weight of each bin, proportional to the number of particles of the simulation which were histogrammed into the given bin. Finally, to randomly pick a given bin with a probability proportional to this weight, we followed the following procedure. Bins were arranged in the PHSP files in decreasing order of their weight, and a vector with as many elements as bins in the PHSP (of the order of a few millions) was built. Each element of the vector would contain the (normalized to one) accumulated probability (weight) from all the previous bins. To select a bin, a random number from 0 to 1 was picked. Then the index of the bin containing the minimum accumulated probability which was larger than this random number was found with a bisection algorithm.

Overall, the debinning process required the calculation of two random numbers per particle, and some

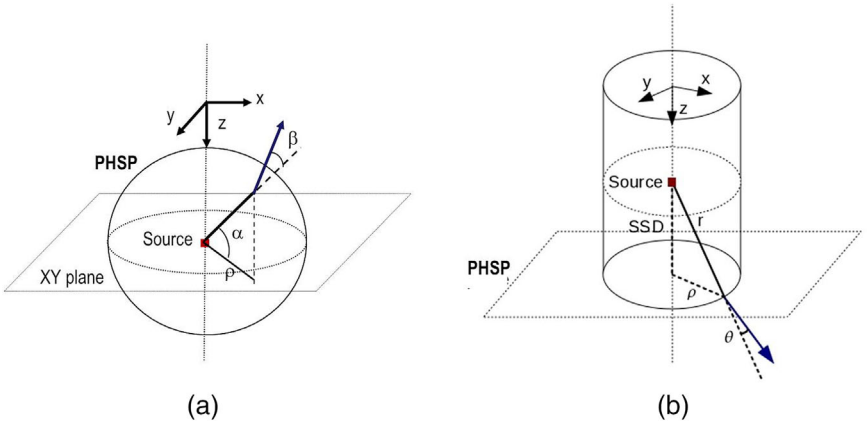

FIGURE 3 Schematic representation of the PHSP parameterization for (a) needle and spherical applicators and (b) flat and surface applicators

operations. In the case of the WC-MC algorithm, with a small number of additional operations per history, debinning required almost as many operations (and random number extractions) as the actual calculation of the dose deposited by every particle, so it represented a substantial fraction in the computation time of the simulations for this algorithm, being however, a minor contribution of the total time spent in the case of the HMC.

2.11 | Woodcock tracking algorithm

Woodcock algorithm[29] was used to track the particles in the WC-MC version of the code and the secondary particles in the HMC version. This algorithm is very efficient in GPU, reaching an acceleration of around one order of magnitude when compared to other tracking algorithms.[18]

Among the advantages of the algorithm is that virtual interactions, which do not update the map of deposited energy, help to hide the latency of much slower global memory update. Briefly described, when a thread of the GPU needs to access global memory, as this is a very slow operation, other thread takes over until the memory access is ready. Eventually, the WC-MC is bounded by global memory accesses and atomic instructions, and the penalty for executing virtual interactions is negligible.

2.12 | Random numbers

One of the trickiest points of the GPU implementation was the random number generation, which, in the end, represented a very significant part of the computing time. There are several random generator routines in the literature that can be used to do this.[35–37] For this work, the pseudo-random generator routine from PENELOPE called RANECU[20,38] was chosen and implemented in GPU. This algorithm has proven to be very fast and to produce random numbers very suitable for MC codes. However, we still had to ensure that the thousands of threads employed in the computations produced random numbers independent among them. Mathematically rigorous methods to parallelize RANECU had been described and used in other MC codes.[39] In our case, several millions of random seeds were precomputed so that each individual thread could use different initial seeds. Each pair of seeds provided an independent sequence of random numbers for each thread. Further, random numbers were shuffled periodically by random intervals in each thread, to keep the independence of random number sequences among different histories.

2.13 | Phase space generation

The XIORT-MC algorithms extract the information of every particle from external PHSP files. These PHSP files were previously obtained with detailed MC simulations for all INTRABEAM applicators. PenEasy,[40] a program designed for PENELOPE-2008, was used for the simulations. For the needle and spherical applicators, we simulated a quasi-punctual source of photons emitting isotropically from the center of the applicators and interacting with a standard geometry per each applicator size. For flat and surface applicators, we used the vendor specifications to accurately reproduce all the geometries. The X-rays energy spectrum used in the simulations was obtained from a MC characterization of the INTRABEAM X-ray source described in Ref. 28. A total of 10^9 particles were stored in the PHSP files of every applicator and binned following the parameterization described in section 2.10.

2.14 | Performance study

We compared the execution time to assess the speed-up factor reached with the CPU and GPU versions of the HMC and the WC-MC against the reference MC penEasy. To do this, we evaluated the XIORT-MC algorithms against codes written both in CPU and in GPU. For the CPU simulations, one core of a CPU of an Intel Xeon W-2155 @ 3.30 GHz was used. Although the

TABLE 1 Specifications of the GPU cards used in this study

	GTX 1080 Ti	RTX 3090
Date	February 2017	September 2020
Architecture	Pascal (6.1)	Ampere (8.5)
Memory	11 GB	24 GB
Cores	3584	10496
TFLOPS	11.3	35.6
Memory Bandwidth (GB/s)	484	936

simulations were carried out on a computer cluster, it is worth pointing out that each simulation ran on a single CPU core. On multiple core CPU, one may expect a speed-up factor of at most the number of available cores. As for the GPUs, we compared the GPU-HMC with the WC-MC in a NVIDIA GeForce GTX 1080 Ti with 3584 cores and maximum clock rate of 1.58 GHz (GPU1) and in a NVIDIA GeForce RTX 3090 with 10496 cores and maximum clock rate of 1.7 GHz (GPU2). Some specifications of both GPUs appear in Table 1. The CPU and GPU versions of the codes were very similar, the main difference is that in the CPU versions, the calculation of every m-history was sequential, one after the previous one, while in the GPU thousands of histories were computed simultaneously.

2.15 | Statistical uncertainties and efficiency enhancement

A detailed study regarding the relative average uncertainty was performed. The HMC and the WC-MC were tested in the water phantom and in three CT-based simulations described in the following section. The number of particles and the simulation time needed to reach a level of relative average uncertainty below 5% was studied, and the efficiency of each simulation was obtained.

To do this, 50 different simulations of each case were performed to ensure enough independent data. The standard deviation was obtained in each voxel from those 50 simulations.

Statistical uncertainty in a voxel j, $\sigma_D(j)$, was defined as Equation (6), where N is the number of independent simulations, $D_i(j)$ is the dose in the voxel j of each simulation I, and $\mu(j)$ is the mean value of the dose in voxel j.

$$\sigma_D(j) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(D_i(j) - \mu(j)\right)^2}. \quad (6)$$

The relative average uncertainty σ_D was defined as the mean value of the ratio between the statistical uncertainty in the voxel j, $\sigma_D(j)$, and the mean dose $\mu(j)$ obtained from voxels with dose above 5% of the maximum dose, $N_{D>0.05Dmax}$.

$$\sigma_D = \frac{1}{N_{D>0.05D_{max}}}\sum_{D>0.05D_{max}}\frac{\sigma_D(j)}{\mu(j)}. \quad (7)$$

This approach of studying the uncertainty was chosen instead of the history-by-history approach[20,41] because, when normalization is applied, uncertainties due to sampling are almost completely removed, so the history-to-history approach would no longer reflect the actual uncertainty of the normalized dose. In the case that no-normalization is employed, we have verified that the estimates for relative average uncertainty agree with the ones of the history-by-history approach.

The efficiency[41] was expressed as Equation (8), where σ_D^2 is the relative average uncertainty and T is the simulation time.

$$\varepsilon = \frac{1}{\sigma_D^2 T}. \quad (8)$$

2.16 | Dosimetric evaluation

The XIORT-MC was tested for accuracy in clinical conditions. To this end, dose distributions were computed with the HMC and with the WC-MC in different scenarios and compared to a reference MC algorithm. PenEasy[40] was chosen as the gold standard to compare the results of our codes.

The test, cases were calculated for all INTRABEAM applicators and included a homogeneous water phantom, a water-bone phantom consisting of a 2-mm thick layer of bone located inside a water phantom at 5 mm depth and a water-lung phantom consisting of a 10-mm thick layer of lung inside a water phantom at the same depth. For flat and surface applicators, the applicator was located normal to the phantom surface, while for spherical and needle applicators, the phantoms were designed surrounding the applicator. Voxelized phantoms with dimensions of (401, 401, 401) voxels were used for the simulations, and the voxel size was set to (0.25, 0.25, 0.25) mm^3.

Three CT-based simulations were also included in the validation. A breast CT was used to simulate a partial breast irradiation close to a rib with a 4 cm diameter spherical applicator, a head CT was used to simulate a sarcoma treatment with a 1 cm diameter surface applicator in the nose, where air inside the nasal cavity might affect the dose distribution, and another head CT was used to simulate a treatment of a brain tumor with the needle applicator. The CTs were segmented in seven materials and the densities were derived, following the method described by Schneider et al.[42] The voxel size was also (0.25, 0.25, 0.25) mm^3.

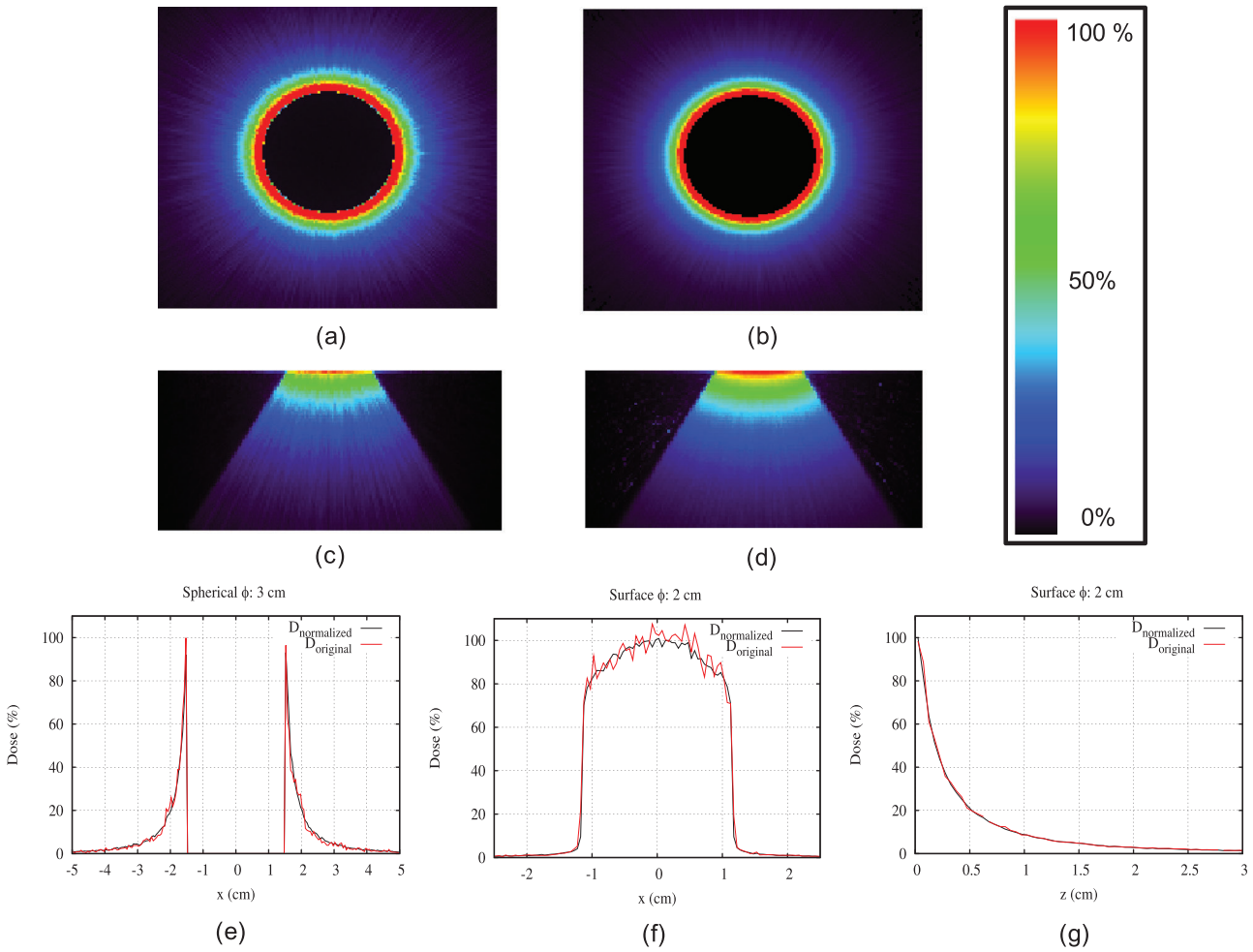

FIGURE 4 Effect of the normalization on dose distributions in water for a 3 cm diameter spherical applicator (top) and for a 2 cm diameter surface applicator (bottom) calculated with the HMC with 5 million m-histories. (a) and (c) are the original doses and (b) and (d) are the normalized doses. Transverse dose profiles calculated along (e) the central axis of the 3 cm diameter spherical applicator, (f) at 5 mm depth and (g) along the vertical axis for the 2 cm diameter surface applicator

The energy absorption was set to 1 keV for photons and 50 keV for electrons in penEasy, to guarantee that no electrons were tracked in the reference simulations. The average angular deflection (C1) and the maximum average energy loss (C2) were set to 0.05. The number of histories used in each simulation was chosen so that $\sigma_D = 5\%$.

A gamma index[34] was used to compare dose volumes obtained with the full detailed MC simulation and XIORT-MC algorithms. In treatment plans consisting of entirely steep gradients (brachytherapy, stereotactic radiosurgery, IORT), the use of similar tolerance in dose and distance is misleading, since most points will fail on distance not dose (magnitude). Therefore, asymmetric tolerances of 7% in dose difference and 0.5 mm in distance-to-agreement were chosen to compute the gamma for the steep gradient of INTRABEAM dose distributions.[3,11,28,43] A solution would be accepted if at least 95% of the voxels pass the gamma evaluation with respect to the reference dose, with a threshold set at 5% of the maximum dose D_{max}.[33]

3 | RESULTS

3.1 | Effect of the normalization

Figure 4 shows the effect of the normalization of dose distributions in water for a 3 cm diameter spherical applicator and for a 2 cm diameter surface applicator with 5 million initial m-histories. Dose profiles have been extracted and compared too. When the normalization is applied, dose distributions show uniformity and no artifacts are visible. The efficiency ratios between normalized and not normalized dose distributions in water for all INTRABEAM applicators are presented in Figure 5. These results have been calculated in a NVIDIA GeForce GTX 1080 Ti GPU card. Similar results have

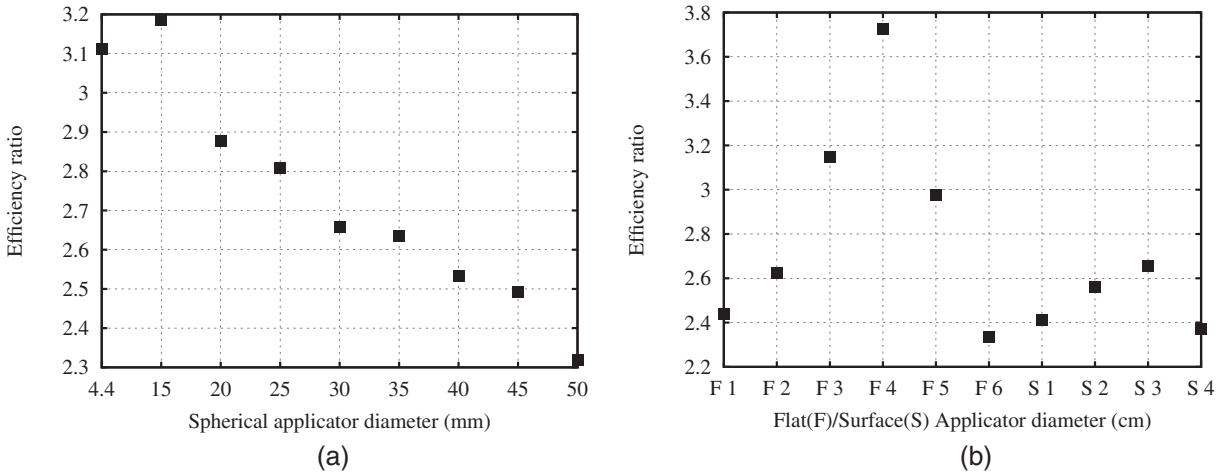

FIGURE 5 Efficiency ratios between normalized and not normalized dose distributions in water for all INTRABEAM spherical (a) and flat/surface (b) applicators calculated in a NVIDIA GeForce GTX 1080 Ti

TABLE 2 7%-0.5 mm 3D gamma index evaluation between the reference MC penEasy and the normalized dose from the HMC and the WC-MC for INTRABEAM needle and spherical applicators for the test cases evaluated (5% threshold)

Spherical applicators										
Phantom	**Code**	**4.4 mm**	**15 mm**	**20 Mm**	**25 mm**	**30 mm**	**35 mm**	**40 mm**	**45 mm**	**50 mm**
Water	HMC	98.3	98.6	98.7	98.6	98.5	98.4	98.4	98.3	98.0
	WC-MC	99.2	98.9	98.8	98.8	98.7	98.8	98.9	99.1	99.2
Water-bone	HMC	96.4	97.4	97.4	97.5	97.4	97.5	97.3	97.4	96.9
	WC-MC	98.9	97.8	97.5	97.4	97.4	97.5	97.7	97.9	98.2
Water-lung	HMC	96.8	99.6	99.8	99.9	99.9	99.9	99.8	99.9	99.8
	WC-MC	99.1	99.1	99.0	99.1	99.1	99.2	99.4	99.6	99.8

been obtained for a NVIDIA GeForce RTX 3090. The results show an average enhancement on the efficiency of 2.75 with the incorporation of the normalization on the dose.

3.2 | Dosimetric evaluation

3.2.1 | Homogeneous and heterogeneous phantoms

Table 2 shows the results of the 7%-0.5 mm gamma evaluation of the HMC and the WC-MC simulations against the reference MC penEasy for the needle and spherical applicators in the water, water-bone and water-lung phantoms. The same information is shown in Table 3 for flat and surface applicators. For all the cases, a good agreement was achieved with more than 95% voxels fulfilling the 3D gamma index evaluation. Figure 6 shows the comparison of the depth dose profiles in the first centimeter of the water-bone phantom simulated for one of each applicator type with the MC-code penEasy (solid lines) and with the HMC (dashed lines). An intermediate size of 3 cm has been chosen for this representation.

3.2.2 | CT-based simulations

Figure 7 shows the dose distributions obtained with the HMC and with the reference MC penEasy code for nose sarcoma, partial breast irradiation, and brain tumor simulations. A good agreement was also obtained in these simulations. The percentage of voxels fulfilling the 7%-0.5 mm gamma evaluation for the HMC and the WC-MC against the reference MC penEasy is presented in Table 4.

3.3 | Performance study

Figure 8 shows the results of the performance study done to all INTRABEAM spherical applicators in water to check that the acceleration reached with the HMC and with the WC-MC, using both CPU and GPU implementations. The same results are shown for all flat and surface

TABLE 3 7%-0.5 mm 3D gamma index evaluation between the reference MC penEasy and the normalized dose from the HMC and the WC-MC for INTRABEAM flat and surface applicators for the test cases evaluated (5% threshold)

		Flat applicators						Surface applicators			
Phantom	**Code**	**10 mm**	**20 mm**	**30 mm**	**40 mm**	**50 mm**	**60 mm**	**10 mm**	**20 mm**	**30 mm**	**40 mm**
Water	HMC	100.0	100.0	100.0	99.9	99.9	99.8	100.0	100.0	99.7	99.8
	WC-MC	99.3	99.7	99.8	99.8	99.8	99.9	99.6	99.8	99.9	99.9
Water-bone	HMC	99.9	99.5	99.7	99.3	99.2	98.9	100.0	99.7	96.7	98.3
	WC-MC	96.5	98.9	99.2	99.3	99.5	99.6	97.3	98.7	97.5	99.3
Water-lung	HMC	100.0	100.0	100.0	99.9	99.8	99.6	100.0	100.0	99.8	99.7
	WC-MC	100.0	100.0	100.0	100.0	99.9	99.9	100.0	100.0	100.0	100.0

TABLE 4 Performance comparisons for the CT-based simulations

Treatment type	**Code**	**Number of particles**	**CPU/GPU**	**Simulation time (s)**	**Efficiency (s^{-1})**	**Speed factor with respect to penEasy**	**7%-5 mm gamma with penEasy**
Sarcoma	penEasy	$5.7{\cdot}10^{8}$	CPU	$3.1{\cdot}10^{4}$	$1.3{\cdot}10^{-2}$	–	–
	WC-MC	$6{\cdot}10^{8}$	CPU	306	1.3	100.0	99.9
			GPU 1	2.6	152.0	$1.2{\cdot}10^{4}$	
			GPU 2	0.8	495	$3.8{\cdot}10^{4}$	
	HMC	$1.1{\cdot}10^{6}$	CPU	33.5	11.9	915.4	99.1
			GPU 1	2.1	188.2	$1.5{\cdot}10^{4}$	
			GPU 2	0.2	2230.0	$1.7{\cdot}10^{5}$	
Breast	penEasy	$1.8{\cdot}10^{9}$	CPU	$9.0{\cdot}10^{4}$	$4.5{\cdot}10^{-3}$	–	–
	WC-MC	$2.0{\cdot}10^{9}$	CPU	$2{\cdot}10^{3}$	0.2	44.4	99.6
			GPU 1	24.4	17.4	$3.9{\cdot}10^{3}$	
			GPU 2	4.3	93.6	$2.1{\cdot}10^{4}$	
	HMC	$6.5{\cdot}10^{6}$	CPU	781.0	0.5	116.3	99.4
			GPU 1	17.4	23.0	$5.1{\cdot}10^{3}$	
			GPU 2	0.5	737.0	$1.6{\cdot}10^{5}$	
Brain	penEasy	$1.4{\cdot}10^{7}$	CPU	770.0	$5.4{\cdot}10^{-1}$	–	–
	WC-MC	$1.5{\cdot}10^{7}$	CPU	12.7	32.9	60.9	99.2
			GPU 1	1.1	347.1	642.8	
			GPU 2	0.04	$1.1{\cdot}10^{4}$	$2.0{\cdot}10^{4}$	
	HMC	$2{\cdot}10^{5}$	CPU	12.7	32.9	60.9	98.7
			GPU 1	1.3	321.7	595.7	
			GPU 2	0.02	$2.3{\cdot}10^{4}$	$4.3{\cdot}10^{4}$	

CPU: Intel Xeon W-2155. GPU 1: NVIDIA GeForce GTX 1080 Ti. GPU2: NVIDIA GeForce RTX 3090.
The results include total number of particles, simulation time, efficiency, speed factor, and gamma evaluation (7%-0.5 mm with 5% threshold) for the CPU and GPU implementations of the HMC and the WC-MC and for the reference MC penEasy. The number of particles of each simulation has been chosen to reach $\sigma_D = 5\%$. Two GPU cards have been studied.

applicators in Figure 9. The HMC and the WC-MC are comparable in terms of efficiency, although the HMC is slightly more efficient overall. If we compare the CPU implementation of both codes against the reference MC penEasy, the XIORT-MC algorithms are much more efficient.

Table 4 shows the results of the performance study in the three CT-based simulations. These results include total number of particles, simulation time, efficiency, speed factor, and gamma evaluation for the CPU and GPU implementations of the HMC and the WC-MC, and for the reference MC penEasy. The total number of particles of each simulation has been chosen to reach $\sigma_D = 5\%$.

4 | DISCUSSION

In this article, we have described XIORT-MC, a dose calculation tool for INTRABEAM that has been tested for all

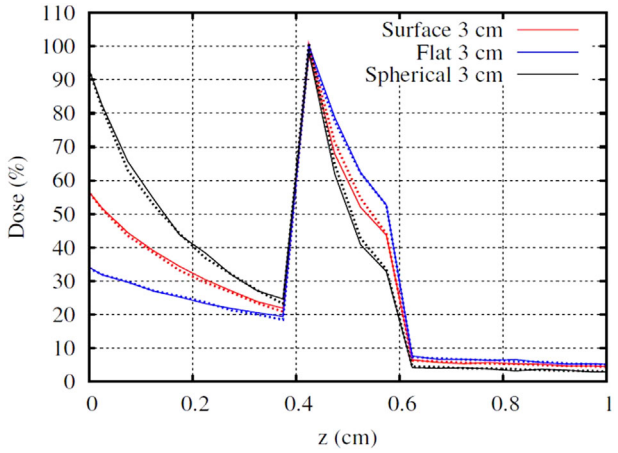

FIGURE 6 Comparison of the first centimeter of the depth dose in the water-bone heterogeneous phantom profiles for a 3 cm diameter surface, flat and spherical applicator, simulated with the reference MC penEasy and with the HMC. Solid lines represent the dose calculated with the MC simulation and the dashed lines the dose calculated with the HMC

commercial applicators. It takes into consideration photoelectric, Rayleigh and Compton interactions. Physics from PENELOPE[20] was incorporated for the simulation of Compton and Rayleigh interactions. Tables with the attenuation coefficients for the different materials were also extracted from the PENELOPE database. The user can select the dose computation algorithm and the CPU or GPU execution. Two algorithms have been incorporated in XIORT-MC: HMC and WC-MC.

The HMC algorithm has been described in detail in this work for the first time. Several novel extreme variance reduction techniques were incorporated to speed up calculations. The particles were treated as m-histories and studied macroscopically. After each step, the m-history was forced to interact, and its weight was updated. A normalization of the dose was also introduced to reduce the stochastic noise and decrease the number of initial particles required. The effect of the normalization in the dose and uncertainty has been studied and an evaluation of the reduction of the statistical noise in normalized and not normalized dose distributions has been done for all INTRABEAM applicators, resulting in an enhancement on the efficiency of an average factor of 2.7 with the incorporation of the normalization on the dose.

The XIORT-MC algorithms were validated in homogeneous and heterogeneous phantoms for all commercial INTRABEAM applicators, as well as in three CT-based simulations. A good agreement was reached in the two codes for all the situations studied, with more than 95% voxels fulfilling a 7%-0.5 mm 3D gamma evaluation in all cases.

MC methods are, in principle, ideally suitable to run on the GPU, where we have every history or m-history to be run by an independent thread. In a regular MC, histories are independent, one of another and, in the case of photons, as it is the case in the WC-MC, each history would contain a very small number of true interactions. That is, the number of atomic updates required to score the dose image, per history (and then in each thread), is relatively small and a good balance of number of operations to atomic instructions is achieved, which allows to hide the latency to slow accesses to global memory. However, in the GPU-HMC, the number of operations and memory updates (atomic global memory writes) per m-history, hence per thread, is much larger than for the GPU-WC-MC. Indeed, each m-history in the GPU-HMC, computed by an independent thread, contains thousands of atomic updates to the dose map and the normalization histogram. In the HMC, the m-histories are forced to interact after each step of the particle, and in each interaction, the deposited energy and the number of interaction (for the normalization procedure) need to be stored with atomic writing. In the case of the WC-MC, the particle typically travels several voxels without interacting. And when it does interact, only the deposited dose is stored.

On the other hand, the GPU-HMC algorithm will keep coherence of the thread to a much better level than the WC-MC, with every thread executing the same instructions at the same time, as no branches are needed to decide the fate of m-histories, contrarily to the WC algorithms.

The number of updates implies a modest performance penalty for a single thread execution on a single core of a CPU, but it is a very limiting factor for the performance of the GPU-HMC implementation or a parallel multithread execution on a multicore CPU, which may lead to the algorithm to be memory bound, that is, its performance being limited by the memory bandwidth. Actually, the GPU-WC-MC tracking is up to 50 times faster than the GPU-HMC for spherical applicators and up to 100 times faster for the flat and surface applicators.

Nevertheless, thanks to dose normalization and other variance reduction properties of the algorithm, the GPU-HMC is as efficient as the conventional GPU-based MC algorithms, even more efficient when high gradients in dose deposition or high-density tissues are involved, so the number of histories needed in the HMC to reach the same level of noise as a regular MC is typically 250 times smaller for spherical applicators and 400 times smaller for flat and surface applicators.

The decoding of the PHSPs inside the XIORT-MC, a process that requires a significant number of operations, may contribute remarkably to the total computation time when a large number of histories is needed (WC-MC). Particularly, the required number of histories in the GPU-WC-MC for the same relative average uncertainty is much higher than for the GPU-HMC, so a very significant percentage of the simulation time in the GPU-WC-MC is due to the debinning. Around 70% of the

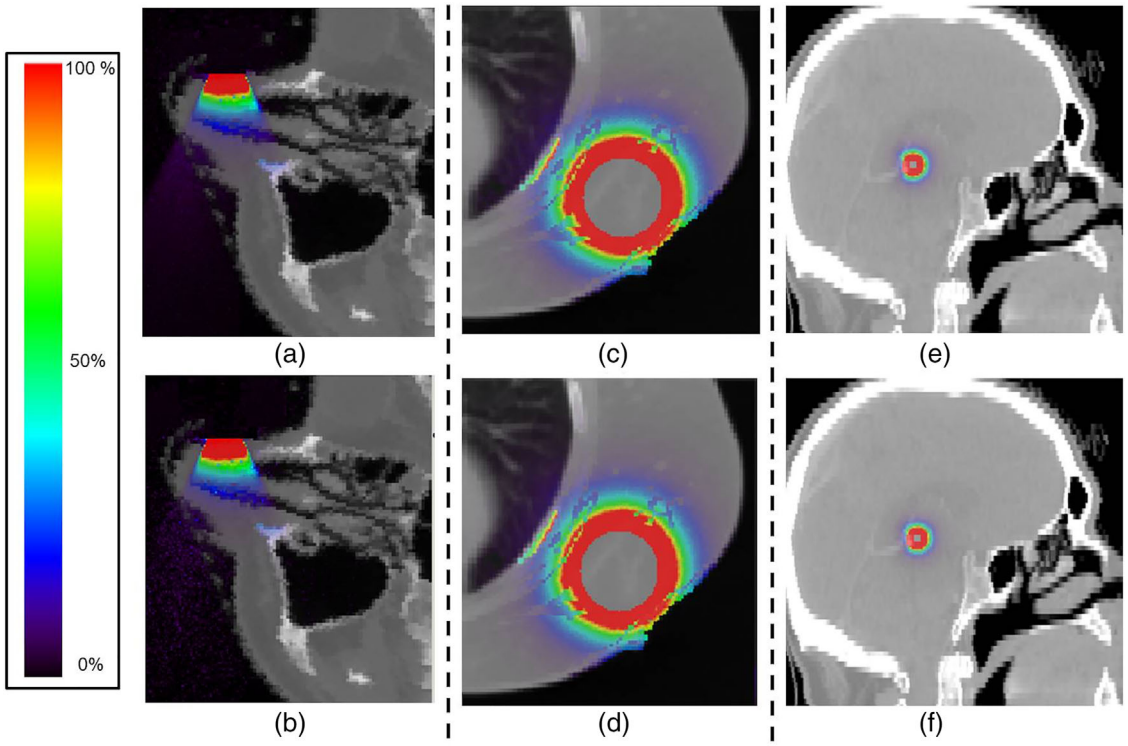

FIGURE 7 Left: Dose comparison for a simulation of a sarcoma treatment in the nose with a 1 cm diameter surface applicator calculated with the HMC (a) and with the reference MC penEasy (b) Center: Simulation of a partial breast irradiation with a 4 cm diameter spherical applicator calculated with the HMC (c) and with penEasy (d) Right: Simulation of a brain tumor treatment with the needle applicator calculated with the HMC (e) and with penEasy (f)

simulation time in the GPU-WC-MC is spent into the debinning process, while in the case of the GPU-HMC, it is less than 20%. The relative importance of the debinning piece of code for the WC-MC in the GPU execution time is essentially due to the fact that for each history, after debinning, there is really little relevant to compute, as at these energies, a majority of photons interacts only once and are absorbed.

Therefore, on the one hand, the GPU-HMC presents a much larger number of atomic updates and random number generator callings than the GPU-WC-MC per m-history, while on the other hand, the GPU-WC-MC requires the debinning of a much larger number of particles than the HMC. All put together, both GPU-HMC and GPU-WC-MC are comparable in terms of efficiency. The efficiency study in water shows that, for the less powerful GPU (GPU1), the GPU-HMC is slightly less efficient than the GPU-WC-MC for the smaller applicators, where fewer histories are required, and more efficient for the larger ones, where the number of histories required to reach an acceptable level of noise is much larger and normalization plays a major role in the uncertainty reduction. When the calculations are performed in the more powerful GPU card (GPU2), the GPU-HMC is more efficient for all applicators than the GPU-WC-MC. In average, the GPU-HMC is about 1.4 times more efficient than the GPU-WC-MC in the GPU1 and about seven times more efficient when running in the GPU2 for the spherical applicators, while for the flat and surface applicators these ratios are about 1.2 in GPU1 and 4.5 in GPU2. We can trace back this performance difference to: (i) the relative importance of the time devoted to debinning the histories in one code and the other. (ii) The number of updates of the dose image per second seems to saturate to a very similar value for both algorithms for the same level of noise, and it is limited by the memory bandwidth of the GPUs, which is much larger for GPU2. The most powerful card significantly speeds up the algorithms by a factor up to 70 with respect to the 1080 GTX.

Regarding the CPU implementation of the codes, both CPU-HMC and CPU-WC-MC are much more efficient than the reference MC penEasy. In the case of spherical applicators, the CPU-HMC is about 95 times more efficient than penEasy and the WC-MC is about 55 times more efficient. For flat and surface applicators, the CPU-HMC is 105 times faster than penEasy, and 49 times more efficient in the case of the CPU-WC-MC. If we compare, the efficiency reached with the CPU version of the HMC and the WC-MC, we can see that the CPU-HMC is slightly more efficient than the CPU-WC-MC, about 1.7 times more efficient for the spherical

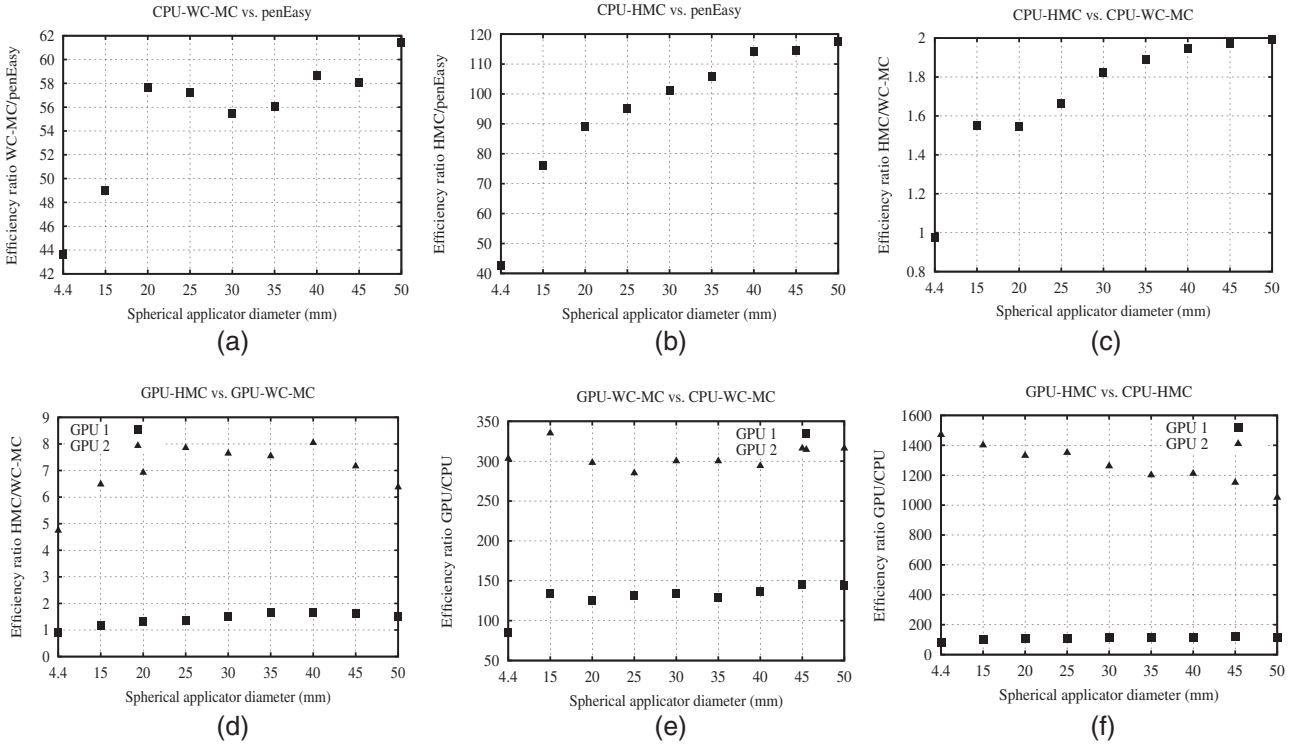

FIGURE 8 Efficiency ratios for all INTRABEAM spherical applicators for dose distributions in water calculated with the CPU-WC-MC versus penEasy (a), the CPU-HMC versus penEasy (b), the CPU-HMC versus CPU-WC-MC (c), the GPU-HMC versus the GPU-WC-MC (d), the GPU-WC-MC versus the CPU-WC-MC, (e) and the GPU-HMC versus the CPU-HMC (f). For the GPU comparisons, simulations have been made in a NVIDIA GeForce GTX 1080 Ti (GPU1) and in a NVIDIA GeForce RTX 3090 (GPU2)

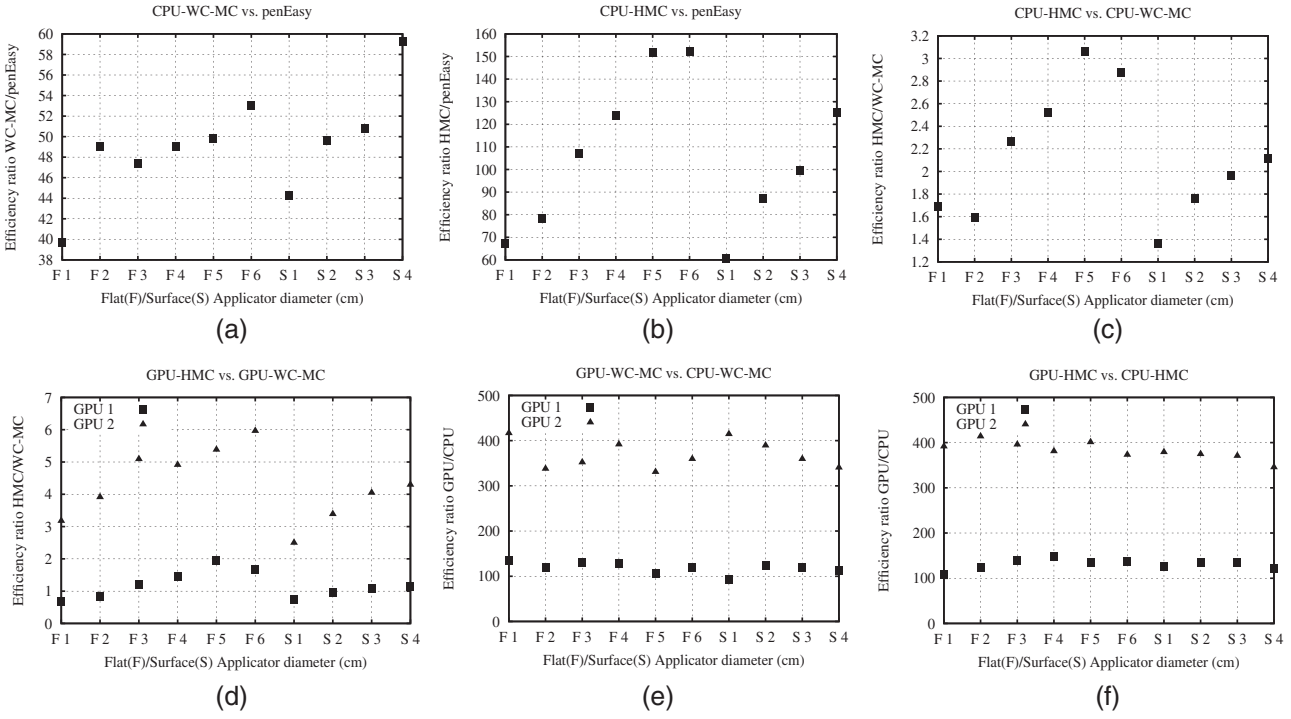

FIGURE 9 Efficiency ratios for all INTRABEAM flat and surface applicators for dose distributions in water calculated with the CPU-WC-MC versus penEasy (a), the CPU-HMC versus penEasy (b), the CPU-HMC versus CPU-WC-MC (c), the GPU-HMC versus the GPU-WC-MC (d), the GPU-WC-MC versus the CPU-WC-MC (e) and, the GPU-HMC vs. the CPU-HMC (f). For the GPU comparisons, simulations have been made in a NVIDIA GeForce GTX 1080 Ti (GPU1) and in a NVIDIA GeForce RTX 3090 (GPU2)

applicators and about 2.1 times more efficient for the flat and surface applicators.

The INTRABEAM dose distributions present very high gradients, which makes any MC simulation very inefficient. Indeed, the number of particles needed to calculate dose volumes with a reasonable level of statistical noise for INTRABEAM applicators is very high, and consequently, the simulation time with any pure MC algorithm running in a CPU will be extended up to several days. The CPU version of the HMC was developed as an alternative to the MC codes running in CPU and dose distributions with the accuracy of a MC simulation and with an acceptable level of noise were obtained in less than 20 min in one core of a CPU. As multicore CPUs with more than 20 cores are common nowadays, calculations below 1 min are easily achieved in modern CPUs.

Further accelerations can be reached with GPUs. Several works presented speed up MC calculations with GPUs. Codes, such as MC-GPU,[18] GPU DPM,[21] GPUMCD,[14] MCX-CL,[22] and ARCHER$_{RT}$,[23] reach a very remarkable acceleration factor when compared to any CPU-based MC. However, most of them are not optimized for working in the INTRABEAM energy, which range from 1 to 50 keV. The XIORT-MC tool presented in this work solves these difficulties. The user can choose between a MC algorithm, the WC-MC in GPU, or a hybrid MC in GPU, which has proven to be more efficient for the larger applicators and obtain accurate dose distributions in less than a second. Specifically, in the cases studied in this work, dose distributions with a noise level below 5% have been obtained for the breast cancer treatment in 0.5 s with the HMC, in 0.2 s for the nose sarcoma, and in 0.02 s for the brain tumor treatment, making it possible to obtain real-time dose distributions as needed in a treatment plan for the INTRABEAM. These numbers are slightly higher for the WC-MC algorithm, but still dose distributions have been obtained in 4.3 s for the breast simulation, in 0.8 s for the sarcoma, and in 0.04 s for the brain simulation.

A recent study by Bosman et al.[44] showed a 2500 acceleration factor when calculating a dose distribution for kilovoltage X-rays with the MC-GPU in a NVIDIA GeForce 1080Ti 11GB GPU and comparing it to penEasy, same as GPU1 we used for the timing study. In our case, dose with the GPU1-WC-MC was obtained 3870 times faster than penEasy for the breast CT and 5100 times faster if we used the GPU1-HMC.

In terms of performance, the HMC code in a 20-cores CPU will be equivalent to GPU1, while more than 500 (ideal) cores will be needed to compete with GPU2. Cost and power considerations, however, are thus, in favor of the GPU implementations. Further, while memory addressing in modern CPUs and standard computers would allow to easily handle 128 GB of RAM or more, making it possible to run typically 40 independent instances of the MC code, with their own image dose space, in a multicore CPU without the need for atomic memory updates, single-node 500 cores solution are far from being common or inexpensive. And most importantly, the computation capabilities and amount of memory available in GPUs both as per unit of consumed power as per purchase cost are growing faster than the ones of the CPU, making the GPU solution more sustainable in the medium range.

As said before, our strategy to port to GPU, the dose calculation codes was to obtain a code with the least possible transformations, with regards to the CPU version. A more aggressive strategy to optimize the code fully redesigning the algorithms to fit into a given GPU may introduce further speed gains and, in this sense, the simplest algorithm WC-MC would be easier targeted for deeper optimization. But the speed gains already obtained after this straightforward port to GPU were already substantial enough to make a strong case in favor of GPU.

Both codes have proved to perform well enough for INTRABEAM dose planning. The HMC is faster than the WC-MC, while the WC-MC is slightly more accurate than the HMC. In clinical practice, the decision of choosing between one code or another may depend on each specific case. However, the authors believe that, in general, the HMC is a better tool for INTRABEAM because of the noise reduction on the dose distributions. Furthermore, when introduced into the daily planning, to be able to calculate dose distributions in less than 0.5 s, may allow to replan as many times as necessary or even to generate a broad set with different treatment options and select the optimal treatment in an amenable time. It will also facilitate the development of inverse planning tools.

5 | CONCLUSIONS

The XIORT-MC described in this work is a powerful tool for treatment planning with the INTRABEAM.[24–28,45] The user can select between CPU or GPU execution and between a MC or a HMC algorithm while keeping a high-dose prediction accuracy. The CPU-based version of the code already reaches a speed-up factor of about 50-100 when compared to a regular MC. The GPU implementation has further increased this speed. Despite the particle tracking of the WC-MC being faster than the HMC, the extraction of the particle information from the PHSP file takes a large fraction of the time, and thanks to the variance reduction techniques implemented in the HMC, up to 400 times fewer particles are needed in this algorithm to reach the same level of noise than the WC-MC. Therefore, dose with noise below 5% has been obtained in realistic situations in less than 5 s with the WC-MC and in less than 0.5 s with the HMC, making possible real-time dose computation for IORT with the INTRABEAM device in clinical conditions. A CPU version of the HMC has already been implemented in *radiance* (GMV, Tres Cantos, Spain), the

first treatment planning system for IORT.[45,46] The GPU version is under evaluation.

ACKNOWLEDGMENTS

Work supported by the Spanish Government (RTI2018-098868-B-I00, RTC-2019-007112-1, XPHASE-LASER), Comunidad de Madrid (B2017/BMD-3888 PRONTO-CM, IND2018/BMD-9990), European Regional Funds and the European Union's Horizon 2020 research and innovation program. This is a contribution for the Moncloa Campus of International Excellence, "Grupo de Física Nuclear-UCM", Ref. 910059. Part of the calculations of this work were performed in the "Clúster de Cálculo para Técnicas Físicas," funded in part by UCM and in part by EU Regional Funds.

CONFLICT OF INTEREST

C. Illana and S. Graullera are employees of GMV, the company that led the development and commercializes Radiance.

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

1. Beatty J, Biggs P, Gall K, et al. A new miniature X-ray device for interstitial radiosurgery: dosimetry. *Med Phys*. 1996;23(1):53-62.
2. Dinsmore M, Harte K, Sliski A, et al. A new miniature X-ray source for interstitial radiosurgery: device description. *Med Phys*. 1996;23(1):45-52.
3. Eaton D, Duck S. Dosimetry measurements with an intra-operative x-ray device. *Phys Med Biol*. 2010;55(12):359.
4. Kraus-Tiefenbacher U, Scheda A, Steil V, et al. Intraoperative radiotherapy (IORT) for breast cancer using the intrabeam™ system. *Tumori J*. 2005;91(4):339-345.
5. Vaidya J, Baum M, Tobias JS, et al. Targeted intra-operative radiotherapy (Targit): an innovative method of treatment for early breast cancer. *Ann Oncol*. 2001;12(8):1075-1080.
6. Giordano FA, Abo-Madyan Y, Brehmer S, et al. Intraoperative radiotherapy (IORT)—a resurrected option for treating glioblastoma. *Translational Can Res*. 2014;3(1):94-105.
7. Schneider F, Greineck F, Clausen S, et al. Development of a novel method for intraoperative radiotherapy during kyphoplasty for spinal metastases (Kypho-IORT). *Int J Radiat Oncol* Biol* Physics*. 2011;81(4):1114-1119.
8. Wenz F, Schneider F, Neumaier C, et al. Kypho-IORT-a novel approach of intraoperative radiotherapy during kyphoplasty for vertebral metastases. *Radiat Oncol*. 2010;5(1):11.
9. Schneider F, Clausen S, Thölking J, Wenz F, Abo-Madyan Y. A novel approach for superficial intraoperative radiotherapy (IORT) using a 50 kV X-ray source: a technical and case report. *J Appl Clin Med Phys*. 2014;15(1):167-176.
10. Bouzid D, Bert J, Dupre P, et al. Monte-Carlo dosimetry for intra-operative radiotherapy using a low energy X-ray source. *Acta Oncol (Madr)*. 2015;54(10):1788-1795.
11. Chiavassa S, Buge F, Hervé C, et al. Monte Carlo evaluation of the effect of inhomogeneities on dose calculation for low energy photons intra-operative radiation therapy in pelvic area. *Physica Med*. 2015;31(8):956-962.
12. Hensley FW. Present state and issues in IORT physics. *Radiat Oncol*. 2017;12(1):37.
13. Sempau J, Wilderman SJ, Bielajew AF. DPM, a fast, accurate Monte Carlo code optimized for photon and electron radiotherapy treatment planning dose calculations. *Phys Med Biol*. 2000;45(8):2263.
14. Hissoiny S, Ozell B, Bouchard H, Després P. GPUMCD: a new GPU-oriented Monte Carlo dose calculation platform. *Med Phys*. 2011;38(2):754-764.
15. Tyagi N, Bose A, Chetty IJ. Implementation of the DPM Monte Carlo code on a parallel architecture for treatment planning applications. *Med Phys*. 2004;31(9):2721-2725.
16. Ziegenhein P, Pirner S, Kamerling CP, Oelfke U. Fast CPU-based Monte Carlo simulation for radiotherapy dose calculation. *Phys Med Biol*. 2015;60(15):6097.
17. Pratx G, Xing L. GPU computing in medical physics: a review. *Med Phys*. 2011;38(5):2685-2697.
18. Badal A, Badano A. Accelerating Monte Carlo simulations of photon transport in a voxelized geometry using a massively parallel graphics processing unit. *Med Phys*. 2009;36(11):4878-4880.
19. Baro J, Sempau J, Fernández-Varea J, Salvat F. PENELOPE: an algorithm for Monte Carlo simulation of the penetration and energy loss of electrons and positrons in matter. *Nucl Instrum Methods Phys Res, Sect B*. 1995;100(1):31-46.
20. Salvat F. PENELOPE-2014: a code system for Monte Carlo simulation of electron and photon transport. *Report NEA/NSC/DOC*. Issy-les-Moulineaux, FRA: OECD Nuclear Energy Agency; 2014.
21. Jia X, Gu X, Sempau J, Choi D, Majumdar A, Jiang SB. Development of a GPU-based Monte Carlo dose calculation code for coupled electron–photon transport. *Physics in Medicine & Biology*. 2010;55(11):3077.
22. Yu L, Nina-Paravecino F, Kaeli DR, Fang Q. Scalable and massively parallel Monte Carlo photon transport simulations for heterogeneous computing platforms. *J Biomed Opt*. 2018;23(1):010504.
23. Su L, Yang Y, et al. ARCHERRT–A GPU-based and photon-electron coupled Monte Carlo dose computing engine for radiation therapy: software development and application to helical tomotherapy. *Med Phys*. 2014;41(7):071709.
24. Ibáñez García PB. *Implementation and validation of ultra-fast dosimetric tools for IORT. PhD Thesis*. Universidad Complutense de Madrid; 2018. https://eprints.ucm.es/id/eprint/49186/1/T39908.pdf. Accessed April, 2021.
25. Ibáñez P, Villa-Abaunza A, Hinault P, Pérez N, Udías JM. Abstract ID: 83 Hybrid Monte Carlo for low-energy X-rays intraoperative radiation therapy dose calculation. *Physica Med*. 2017;42:17.
26. Vidal M, Guerra P, Herranz E, Ibáñez P, Udías J. 211: realistic on-the-fly dose calculation for low energy X-rays Intra-Operative Radiation Therapy. *Radiother Oncol*. 2014;110:S103-S104.
27. Vidal M, Ibáñez P, González JC, Guerra P, Udías J. OC-0275: hybrid Monte Carlo dose algorithm for low energy X-rays intraoperative radiation therapy. *Radiother Oncol*. 2014;111:S106-S107.
28. Vidal M, Ibáñez P, Guerra P, et al. Fast optimized Monte Carlo phase-space generation and dose prediction for low energy x-ray intra-operative radiation therapy. *Physics in Medicine & Biology*. 2019;64(7):075002.
29. Woodcock E, Murphy T, Hemmings P, Longworth S, Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. *In Proc. Conf. Applications of Computing Methods to Reactor Problems*. 1965;557(2).
30. Chica U, Anguiano M, Lallena A. Benchmark of PENELOPE for low and medium energy X-rays. *Physica Med*. 2009;25(2):51-57.
31. Croce O, Hachem S, Franchisseur E, Marcie S, Gerard JP, Bordy JM. Contact radiotherapy using a 50 kV X-ray system: evaluation of relative dose distribution with the Monte Carlo code PENELOPE and comparison with measurements. *Radiat Phys Chem*. 2012;81(6):609-617.
32. Ye SJ, Brezovich I, Pareek P, Naqvi SA. Benchmark of PENELOPE code for low-energy photon transport: dose comparisons with MCNP4 and EGS4. *Phys Med Biol*. 2004;49(3):387.

33. Herranz E, Herraiz J, Ibáñez P, et al. Phase space determination from measured dose data for intraoperative electron radiation therapy. *Phys Med Biol*. 2014;60(1):375.
34. Low DA, Harms WB, Mutic S, Purdy JA. A technique for the quantitative evaluation of dose distributions. *Med Phys*. 1998;25(5):656-661.
35. James F. RANLUX: a Fortran implementation of the high-quality pseudorandom number generator of Lüscher. *Comput Phys Commun*. 1994;79(1):111-114.
36. Matsumoto M, Nishimura T. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transact Model Computer Simulation (TOMACS)*. 1998;8(1):3-30.
37. Mascagni M, Srinivasan A. Algorithm 806: sPRNG: a scalable library for pseudorandom number generation. *ACM Transact Math Software (TOMS)*. 2000;26(3):436-461.
38. L'ecuyer P. Efficient and portable combined random number generators. *Commun ACM*. 1988;31(6):742-751.
39. Badal A, Sempau J. A package of Linux scripts for the parallelization of Monte Carlo simulations. *Comput Phys Commun*. 2006;175(6):440-450.
40. Sempau J, Badal A, Brualla L. A penelope-based system for the automated Monte Carlo simulation of clinacs and voxelized geometries-application to far-from-axis fields. *Med Phys*. 2011;38(11):5887-5895.
41. Renaud MA, Roberge D, Seuntjens J. Latent uncertainties of the precalculated track Monte Carlo method. *Med Phys*. 2015;42(1):479-490.
42. Schneider W, Bortfeld T, Schlegel W. Correlation between CT numbers and tissue parameters needed for Monte Carlo simulations of clinical dose distributions. *Phys Med Biol*. 2000;45(2):459.
43. Shamsabadi R, Baghani HR, Mowlavi AA, Azadegan B. Effective energy assessment during breast cancer intraoperative radiotherapy by low-energy X-rays: a Monte Carlo study. *Radiat Environ Biophys*. 2021;60(1):125-134.
44. Bosman DF, Balcaza VG, Delgado C, et al. Validation of the MC-GPU Monte Carlo code against the PENELOPE/penEasy code system and benchmarking against experimental conditions for typical radiation qualities and setups in interventional radiology and cardiology. *Physica Med*. 2021;82:64-71.
45. Valdivieso-Casique MF, Rodríguez R, Rodríguez-Bescós S, et al. RADIANCE—A planning software for intra-operative radiation therapy. *Translat Cancer Res*. 2015;4(2):196-209.
46. Pascau J, Miranda JAS, Calvo FA, et al. An innovative tool for intraoperative electron beam radiotherapy simulation and planning: description and initial evaluation by radiation oncologists. *Int J Radiat Oncol Biol Phys*. 2012;83(2):287-295.

How to cite this article: Ibáñez P, Villa-Abaunza A, Vidal M, et al. XIORT-MC: A real-time MC-based dose computation tool for low-energy X-rays intraoperative radiation therapy. *Med Phys*. 2021;48:8089–8106. https://doi.org/10.1002/mp.15291