Published in final edited form as:

Phys Med Biol. 2017 April 21; 62(8): 3081–3096. doi:10.1088/1361-6560/aa6246.

Accelerated Monte Carlo Simulation on the Chemical Stage in Water Radiolysis using GPU

Zhen Tian, Steve B. Jiang, and Xun Jia

Department of Radiation Oncology, University of Texas Southwestern Medical Center, Dallas, TX 75390

Abstract

The accurate simulation of water radiolysis is an important step to understand the mechanisms of radiobiology and quantitatively test some hypotheses regarding radiobiological effects. However, the simulation of water radiolysis is highly time consuming, taking hours or even days to be completed by a conventional CPU processor. This time limitation hinders cell-level simulations for a number of research studies. We recently initiated efforts to develop gMicroMC, a GPU-based fast microscopic MC simulation package for water radiolysis. The first step of this project focused on accelerating the simulation of the chemical stage, the most time consuming stage in the entire water radiolysis process. A GPU-friendly parallelization strategy was designed to address the highly correlated many-body simulation problem caused by the mutual competitive chemical reactions between the radiolytic molecules. Two cases were tested, using a 750 keV electron and a 5 MeV proton incident in pure water, respectively. The time-dependent yields of all the radiolytic species during the chemical stage were used to evaluate the accuracy of the simulation. The relative differences between our simulation and the Geant4-DNA simulation were on average 5.3% and 4.4% for the two cases. Our package, executed on an Nvidia Titan black GPU card, successfully completed the chemical stage simulation of the two cases within 599.2 s and 489.0 s. As compared with Geant4-DNA that was executed on an Intel i7-5500U CPU processor and needed 28.6 h and 26.8 h for the two cases using a single CPU core, our package achieved a speed-up factor of 171.1-197.2.

1. Introduction

Ionizing radiation causes damage through water radiolysis, with DNA as the primary target of the generated radiolytic molecules (Halliwell and Aruoma, 1991; Hirayama *et al.*, 2009; O'Neill and Wardman, 2009). The accurate modeling of water radiolysis is essential to understand the radiobiological mechanisms and quantitatively test some hypotheses in related problems. For instance, water radiolysis simulation can be used to test two hypotheses, i.e. the interacting radical hypothesis and the oxygen-in-the-track hypothesis, that have been proposed to account for the observation that the increased sensitivity of cellular organisms to irradiation in the presence of oxygen is reduced progressively with increasing linear energy transfer (LET) of radiation (Alper, 1956; Neary, 1965; Alper and Bryant, 1974; Baverstock and Burns, 1981). The first hypothesis proposes that the probability of a fixation reaction occurring between radicals produced along a track is increased as the ion density increases, so the importance of interaction with oxygen gets decreased (Alper, 1956). The second proposes that molecular oxygen is a product of water

radiolysis at high LET and effectively oxygenates cell targets at the time of irradiation (Neary, 1965). Therefore, accurate simulation of water radiolysis is needed for the optimal use of radiation in cancer treatment (Nikjoo *et al.*, 2006).

The radiation action of an ionizing particle in water (water radiolysis) is divided into three consecutive stages. The physical stage occurs within ~ 1 femtosecond after irradiation, and involves transport of radiation particles, energy deposition, and generation of ionized and excited water molecules and subexcitation electrons. In the prechemical stage, typically lasting from ~ 1 femtosecond to ~ 1 picosecond, the radiolytic molecules are formed through the dissociation of ionized and excited water molecules, as well as the solvation of subexcitation electrons. The chemical stage, from ~ 1 picosecond to ~ 1 microsecond, is characterized by diffusion and mutual chemical reactions of the produced radiolytic molecules.

Monte Carlo (MC) simulation is widely recognized as one of the most accurate methods for simulations of water radiolysis (Uehara and Nikjoo, 2006; Kreipl *et al.*, 2009). Efforts have been devoted to developing MC simulation tools to elucidate radiobiological effects. Example of these tools include PARTRAC (Kreipl *et al.*, 2009; Friedland *et al.*, 2011), Geant4-DNA (Incerti *et al.*, 2010; Karamitros *et al.*, 2011; Karamitros *et al.*, 2014; Bernal *et al.*, 2015), and RITRACKS (Plante, 2011b, a). MC simulation of the physical and prechemical stages of an ionizing particle in water is relatively fast, and can be typically completed in seconds or minutes on a CPU. In contrast, the simulation of the chemical stage may take hours or even days. The reasons for the much longer simulation time are multiple. First, the chemical stage involves several orders of magnitude in terms of time, and requires many steps to simulate this dynamic process. Second, an incident ionizing particle can generate a large number of radiolytic molecules in water. The activities of these molecules need to be simulated in this stage. Third, a chemical reaction may occur between two neighboring molecules and multiple molecules may compete for the same reactant. These mutual and competitive reactions cause a highly correlated many-body simulation problem, leading to high algorithm complexity. The requirement for long computational time hinders cell-level MC simulations for multicellular tumors in clinical scenarios (Kalantzis *et al.*, 2012). This limitation indicates the need to accelerate MC simulation of the chemical stage in water radiolysis.

Effort has been devoted to parallelize the simulation of the chemical stage on multi-core CPUs (Kalantzis *et al.*, 2012). Using four and eight CPU cores, speedup factors of ~ 3.5 and ~ 5.7 were achieved, respectively, as compared to using a single core. Because of the recent boom in graphics processing unit (GPU)-based high performance computing, there has been a burst of research developing fast MC simulation tools on this platform for different applications (Jia *et al.*, 2014). Substantial acceleration factors were achieved by GPU-based macroscopic MC dose engines over conventional or multi-threaded CPU-based engines (Jia *et al.*, 2010; Hissoiny *et al.*, 2011; Jia *et al.*, 2012; Tachibana, 2014). GPU is also known to successfully accelerate the simulation of fluid system, contacting particle systems, and cell biological systems (Green, 2010; Gladkov *et al.*, 2011; Dematte, 2012). Inspired by these successful GPU applications, we initiated the development on gMicroMC, a GPU-based microscopic MC simulation package for water radiolysis. The initial development on this

package focused on accelerating simulations of the chemical stage on GPU. We employed a GPU-friendly parallelization strategy to specifically address the highly correlated many-body simulation problem caused by the mutual and competitive reactions between the radiolytic molecules. To our knowledge, this is the first GPU-based simulation package specifically developed to enable fast MC simulation of the chemical stage in water radiolysis.

2. Methods and Materials

2.1 MC Simulation of the Chemical Stage

The chemical stage of water radiolysis consists of two types of chemical kinetics: 1) diffusion of the radiolytic molecules and 2) their mutual chemical reactions. We employed the step-by-step (SBS) method that has been widely used in MC simulations of this stage (Uehara and Nikjoo, 2006; Kreipl *et al.*, 2009; Friedland *et al.*, 2011; Karamitros *et al.*, 2011). In this method, the entire chemical stage was divided into a number of small consecutive time steps. Within each time step, the chemical kinetics of the radiolytic molecules was simulated.

2.1.1 Diffusion Model—Brownian motion was used to model the diffusion of the radiolytic molecules with each being considered as an individual Brownian object with random, independent motion. For a molecule initially located at (x_0, y_0, z_0) , the probability of finding it in an elementary cube $(x + dx, y + dy, z + dz)$ after a time interval of t was given by a 3D Gaussian distribution

$$p(x, y, z)dx dy dz = \frac{1}{(4\pi D\Delta t)^{3/2}} \exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}{4D\Delta t}\right) dx dy dz, \quad (1)$$

where D denotes the diffusion coefficient of the molecule.

2.1.2 Reaction Model—Our package assumed that all chemical reactions were diffusion-controlled, as PARTRAC and Geant4-DNA assumed (Friedland *et al.*, 2011; Karamitros *et al.*, 2014). The criterion to induce a diffusion-controlled reaction was the distance between the two reactants. A reaction would only occur when the distance of the reactants was no greater than its reaction radius. In our package, the radius R of a specific reaction was calculated based on its relationship with the reaction rate constant k , according to the Smoluchowski formalism (Naqvi *et al.*, 1980):

$$k = 4\pi R D_p \left(1 + \frac{R}{\sqrt{\pi D_p \Delta t}}\right) N_A. \quad (2)$$

Here, N_A is the Avogadro constant and Δt denotes the duration of the current time step. $D_p = D_1 + D_2$ is the diffusion coefficient of the reactant pair, which is equal to the sum of the diffusion coefficients of the two reactants D_1 and D_2 .

Once a reaction occurred, the location of the reaction site (x_r, y_r, z_r) was assumed to be located between its two reactants and weighted by their diffusion coefficients; (x_r, y_r, z_r) can be expressed as (Plante, 2011b):

$$(x_r, y_r, z_r) = (x_1, y_1, z_1) \frac{\sqrt{D_2}}{\sqrt{D_1} + \sqrt{D_2}} + (x_2, y_2, z_2) \frac{\sqrt{D_1}}{\sqrt{D_1} + \sqrt{D_2}}, \quad (3)$$

where (x_1, y_1, z_1) and (x_2, y_2, z_2) denote the spatial locations of the two reactants. After the reaction occurred, the reactants disappeared and multiple molecules (zero to three in the reactions considered in this paper) may be produced. These products were first placed at the reaction site. If there existed a potential reactant pair among the products, we placed this pair randomly around the reaction site; the distance between the two was equal to the corresponding reaction radius (Plante, 2011b). This strategy can reduce the likelihood of their immediate recombination.

2.1.3 Dynamic Time Step—In the SBS method, the chemical stage is divided into a number of small consecutive time steps for MC simulation. The sizes of the time steps are essential for the accuracy and efficiency of the simulation. Small time steps ensure that no potential reaction is missed. However, the simulation time of the chemical stage becomes longer because of the increased number of time steps. Because the radiolytic molecules diffuse as time progresses, the frequency of their chemical reactions gradually decreases because of the increased distances between them (Plante, 2011b). This implies that a relatively large step size may be sufficient for the later period of the chemical stage.

To improve efficiency while maintaining accuracy, we employed the dynamic time step strategy that was proposed by Michalik et al. and has been used in Geant4-DNA (Michalik et al., 1998; Karamitros et al., 2014). In this method, given a potential reactant pair, the time interval t_D during which the corresponding reaction would not occur with a 95% confidence interval was calculated as

$$\Delta t_D = \begin{cases} \frac{(d-R)^2}{8(D_1+D_2+2\sqrt{D_1D_2})}, & d > R \\ 0, & d \leq R \end{cases} \quad (4)$$

Here, d denotes the current distance between two potential reactants. The size of the time step, denoted as Δt , was set as the minimum value of t_D between all the reactant pairs. Because all the reactions considered in our simulation were assumed to be diffusion-controlled, a reaction would occur immediately, when the distance between the two potential reactants was no larger than the reaction radius. In this case, both t_D and Δt were zero.

Two issues should be noted regarding this dynamic time step method. First, according to Eq. (2), the reaction radius R used for each time step was dependent on the corresponding step size, which made it difficult to exactly determine t_D . Second, given a potential reactant pair, the value of the corresponding t_D obtained using Eq. (4) would keep decreasing before the reaction finally occurred. This would lead to many steps with gradually reduced step sizes, increasing the overall simulation time and defeating our purpose to employ the dynamic time step strategy. To address these issues, we combined the dynamic step size with the fixed step sizes, a strategy used by PARTRAC (Kreipl *et al.*, 2009). Specifically, we set fixed step size parameters for different time periods in the chemical stage (Table 1). We used the fixed step sizes to calculate the reaction radius R via Eq. (2) and then calculate the actual step size via Eq. (4). They also served as a constraint on the minimum step size allowed in our simulation to ensure efficiency.

2.1.4 Crossing event—Given a potential reactant pair at a certain time step, even if both the pre-diffusion and post-diffusion distances between the two reactants were larger than the reaction radius, the distance may still be smaller at a certain time point during diffusion (Hamm *et al.*, 1998) and reaction may occur. This phenomenon is referred to as the “crossing” event. Hence, we employed Brownian bridge method to estimate the probability of crossing p_c (Karamitros *et al.*, 2014), calculated as

$$p_c = \exp\left(-\frac{(d_{\text{pre}} - R)(d_{\text{post}} - R)}{D_p \Delta t}\right). \quad (5)$$

Here, d_{pre} and d_{post} denote the pre-diffusion and post-diffusion distances between the two reactants for a given time step. In Eq. (5), both d_{pre} and d_{post} should be larger than the reaction radius R . After the diffusion step, a random number uniformly distributed in [0,1] was generated. The crossing event occurred, when the random number was smaller than the calculated probability p_c .

2.1.5 Overall simulation workflow—Based on the aforementioned simulation methods and models, the MC simulation for the chemical stage in water radiolysis can be structured as shown in Table 2.

2.2 Parallelization on GPU

We have implemented the MC simulation workflow on GPU to accelerate the calculation. The flowchart is shown in Fig. 1. The code was initialized by loading the input data and transferring them from CPU to GPU. The simulation looped over the time steps and was terminated when the chemical stage ended. The details of our GPU implementation are presented in the following subsections.

2.2.1 Data Storage on GPU—Two categories of input data were needed for the chemical stage simulation. One was the basic chemistry data to specify the considered chemical kinetics, including 1) types of the radiolytic molecules and the diffusion coefficients of them; 2) reaction equations of the chemical reactions and their reaction rate constants.

Because these basic data were small and kept constant during a simulation, they were stored on the GPU's constant memory for fast access.

Another category was the data of radiolytic molecules after the pre-chemical stage, i.e. their types and spatial locations, which were needed because the current gMicroMC package only supports simulations of the chemical stage. We allocated on the GPU's global memory the following memory spaces. First, a molecule array was allocated to record the evolution of the radiolytic molecules during the simulation. This array was initialized with the input molecule data. Second, a chemical reaction considered in our package may produce at most three products. While two of the products can be recorded at the two reactants' locations in the molecule array, an additional space is needed to record the third product. Hence we allocated a buffer array on GPU's global memory. The size of the buffer array was equal to that of the molecule array. Third, we also allocated a tag array associated with the molecule array on the GPU global memory. The function of this array was to inform the GPU regarding the status of each molecule during simulation. Specifically, a tag value of 0 in this array denoted a molecule that had already undergone a chemical reaction and should be excluded from subsequent simulation (referred to as “a dead molecule”). A value of 1 implied that the molecule had not reacted (referred to as “a live molecule”). A value of 2 referred to a new produced at the current time step. It would not be available for reactions until the next time step (referred to as “a new molecule”). Last, because both the pre-diffusion and post-diffusion distances between two reactants were needed to calculate the crossing-event probability using Eq. (5), a temporary position array was allocated to store the post-diffusion locations of the molecules. The reason to use global memory was that these memory spaces would be updated during simulation and hence the memory should be large enough and support both reading and writing operations.

2.2.2 Search for reactant pairs—The search for potential reactant pairs is an indispensable operation in the simulation. However, this search process is characterized by a high computational complexity. If N molecules are present at a time step, we have to check $N(N-1)/2$ molecule pairs, yielding a computational complexity in the order of N^2 . Hence the search time would be greatly increased for the cases that involve a large number of radiolytic molecules. In our simulation, we assumed that a chemical reaction only occurred when the distance of its reactants was no greater than the reaction radius. With this assumption, we were allowed to search for the potential reactants of a molecule only in its vicinity to reduce code complexity.

To realize this idea, we employed a grid data approach, previously used in GPU-based particle collision simulation of fluids or contacting particle systems (Green, 2010). The basic idea was to divide the volume of interest (VOI), defined as the smallest cube containing all the live radiolytic molecules, into a grid of uniformly sized cells. The cell size was determined by the largest reaction radius out of all the possible reactions. With this grid, for each molecule, we only need to search for its potential reactants within the same cell and the neighboring cells (27 cells in total for 3D grid). A simple 2D example is shown in Fig. 2 to better illustrate this idea. For instance, we only needed to search for potential reactants of the molecule 7 in the corresponding nine shaded cells in Fig 2(a).

Two types of data were needed in this a grid data approach: 1) the ID of a cell containing a given molecule and 2) the IDs of those molecules belonging to a given cell. We employed the following strategies to construct these data. It is worth to note that, because molecule positions varied among time steps, construction of these data was needed at each time step before searching for the potential reactant pairs.

For data type 1), the cell ID was simply determined according to the spatial location of the molecules. However, special attentions were needed to determine data type 2). Specifically, we performed a sorting operation using thrust library to rearrange the molecules with an ascending order of their cell IDs (Bell and Hoberock, 2011). A straightforward approach to identify the molecules for each cell is to store an index array with its length being the number of cells. Each element in this array would denote the index in the sorted molecule array corresponding the first molecule in a cell. Yet this was not feasible on a modern GPU with a limited memory space due to the vast number of cells. Given a typical VOI size of a few cubic millimeters and the reaction radius ranging from sub-nanometers to nanometers, the total number of cells could be $10^{15\sim 21}$. However, it is unnecessary to store the information in this way in light of the fact that the distribution of the radiolytic molecules is usually very inhomogeneous in space and many cells do not contain any molecule. In our implementation, we built a non-empty-cell array A storing the IDs of those non-empty cells and an associated start-position array B . The elements between $B[i]$ to $(B[i + 1] - 1)$ in the sorted molecule array were molecules belonging to the non-empty cell with the ID $A[i]$. Here we followed coding convention that the array index started from 0. As illustrated in Fig.2 (b), for cell 10 corresponding to $A[3]$, the molecules in it were those in the sorted molecule array with indices in the range of 5 ($B[3] = 5$) to 6 ($B[4] - 1 = 6$).

In addition, to search for potential reactants of a molecule within its corresponding 27 candidate cells (9 cells for the 2D example), we also needed the indices of those candidate cells in the non-empty-cell array A . Since the molecules in a given cell required the same information, to avoid repeated calculations, we generated a look-up table. Each row of the table was for one non-empty cell in array A , and each column corresponded to one of the 27 candidate cells (9 cells for the 2D example). An entry value of -1 denoted that the corresponding candidate cell was empty. Take the cell 10 as an example. The corresponding row (4th row) of the look-up table implied that the cells indicated by the element 2, 3, 5, and 6 in array A should be searched, whereas the other five out of nine candidate cells were empty.

2.2.3 Simulation of a time step on GPU—The flowchart is shown in Fig. 1. With the grid data built, a GPU kernel A was launched to determine the step size. In this kernel, each GPU thread was responsible for a live molecule, searching for its potential reactants using the grid data. When a potential reactant pair was identified and their separation was less than the reaction distance, the reaction was simulated. If the reaction did not yield any product, the reactants immediately died with their corresponding tag values set to 0. If the reaction yielded one product, the GPU thread overwrote one of the reactants in the molecule array with the product and its tag value was updated to 2. The other reactant was set to dead (a tag value 0). Similarly, if two products were yielded, both reactants were overwritten by the products. If the simulated reaction had three products, while two reactants were overwritten

by the two products, the third one was written into the buffer array. After all the reactions were simulated, the tag values of the generated products were changed from 2 to 1, so they could participate in reactions afterwards. On the other hand, when a potential reactant pair was identified but the separation between them was larger than the reaction radius, t_D was calculated for the reactant pair. The GPU thread also recorded the minimum t_D value among all the potential reaction pairs involving the molecule that the GPU thread was responsible for. Once kernel *A* was completed, if any reaction occurred, the time step size would be set to 0 and the current time step ended. Otherwise, the thrust library was used to find the minimum t_D among all the threads to determine the actual step size Δt .

After that, a GPU kernel *B* was launched to diffuse the molecules with one thread per molecule. After updating the grid data based on the post-diffusion locations, a GPU kernel *C* was launched. In this kernel, each thread searched for the potential reactants for a live molecule using the updated grid data. For an identified reactant pair, the reaction was simulated, when the separation was no larger than the reaction radius. Otherwise, the thread computed the crossing probability and simulated the reaction if needed. Once the kernel *C* was completed, the current time step ended. We then removed all the dead molecules and moved products in the buffer array into the molecule array to prepare for the next time step.

2.3 Chemical data used in gMicroMC

The default basic chemical data used in our gMicroMC package to specify chemical kinetics are those proposed by PARTRAC, as shown in Tables 3 and 4. The same data were also used in Geant4-DNA. Users are allowed to specify the chemical data for their own applications.

2.4 Testing cases

Two testing cases were studied to validate the accuracy of gMicroMC and to demonstrate its efficiency gain. In case 1, an electron with energy of 750 keV was emitted from the center of a spherical water phantom with a diameter of 3 mm, in which the electron deposited all its energy. In case 2, a proton with energy of 5 MeV was emitted from the center of a spherical water phantom with a diameter of 0.05 mm, in which the proton deposited 0.43 MeV to the phantom. The open source simulation toolkit Geant4-DNA, version 10.2 (Incerti *et al.*, 2010) was used to perform physical and pre-chemical stage simulation of water radiolysis to provide the input data (i.e. the radiolytic molecules generated after the pre-chemical stage) for our chemical stage simulation. The chemical stages of these two cases were also simulated using Geant4-DNA for comparison purposes.

We used the time-dependent radiolytic yield to validate the accuracy of the chemical stage simulation. The yield was defined as the number of radiolytic molecules produced for a deposited energy of 100 eV in the irradiated medium, formulated as

$$G(t) = \frac{N(t)}{E/100\text{eV}}. \quad (6)$$

Here, $N(t)$ denotes the number of molecules for one species type at time t . E , expressed in eV , is the total energy deposited in the medium by the incident ionizing particle. Our obtained yield values were also compared with the values reported in a reference for the 750 keV electron and the 5 MeV proton using the PARTRAC simulation package (Kreipl *et al.*, 2009).

In our study, the Geant4-DNA toolkit was installed on a desktop computer with an Intel i7-5500U CPU card. A single CPU core was used for Geant4-DNA simulation in our study. Our gMicroMC package was executed on an Nvidia GeForce GTX TITAN Black GPU card (Nvidia Corporation, Santa Clara, CA), which had 15 stream multiprocessors and 2880 GPU threads in total. The simulation time of the chemical stage using the Geant4-DNA toolkit and our package were recorded for comparison.

3. Results

Fig. 3 shows the live radiolytic molecules obtained at three different time steps ($t = 1\text{ ps}$, 10^5 ps and 10^6 ps) for the two cases. This figure clearly demonstrated the reaction-diffusion process. This observation also explained why we may use a relatively large step size at the later period of the chemical stage for diffusion-controlled reactions. Besides, it was also observed that the molecules in the proton case had a much denser distribution than those in the electron case because of the high LET of the protons.

3.1 Accuracy

The time-dependent yield values obtained for all the different species in the two cases are shown in Fig. 4 and 5. The yields obtained using Geant4-DNA and those reported in a reference using PARTRAC (Kreipl *et al.*, 2009) are also plotted for comparison purposes. The yields from our simulation were more similar to those obtained by Geant4-DNA, because our package used the radiolytic molecules obtained from Geant4-DNA after the simulation of the physics and prechemical stages as its input data. A relatively large discrepancy was observed between the Geant4-DNA results and the PARTRAC results (directly extracted from Kreipl *et al.*, 2009) in terms of $\cdot\text{OH}$ and e_{aq}^- yields in both cases. This disparity may be explained by the different physical models, cross section data, and dissociative branch parameters used by PARTRAC and Geant4-DNA in their simulation of the physical and prechemical stages. Besides, although the PARTRAC results extracted from the reference were also for a 750 keV electron and a 5 MeV proton, their simulations were conducted using different geometry (a slab of $2.5\text{ }\mu\text{m}$ thick water) and hence different amount of energy deposition, which may also be ascribed to the discrepancy.

The yield values of the radiolytic species present at $1\text{ }\mu\text{s}$ are listed in Table 5. Because of the large amount of the molecules involved in both cases, the statistical uncertainties of these yield values obtained by our package were within 0.1%. The relative differences between our results and the Geant4-DNA results were on average 5.3% and 4.4% for the two cases, respectively. The relative differences achieved for the two species $\cdot\text{OH}$ and e_{aq}^- were within 1.5%. Relatively large differences were found for $\text{H}\cdot$, H_2 , OH^- and H_2O_2 , which was due to their small yield values.

3.2 Efficiency

The simulation times for the two cases are reported in Table 6. Case 1 (750 keV electron) involved 101829 initial radiolytic molecules. The total simulation times were 102865.4 s (~28.6 h) for Geant4-DNA executed on an Intel i7-5500U CPU card (using single CPU thread in our study), and 599.2 s for our package executed on an Nvidia GTX Titan black GPU card. In case 2 (5 MeV proton), there were 56122 initial molecules. It took 96446.5 s (~26.8 h) and 489.0 s for Geant4-DNA and our package, respectively. It was noted that although the number of the initial radiolytic molecules for case 2 was about half of that for case 1, the simulation times of the two cases were comparable in both Geant4-DNA and our package. This should be due to the much denser distribution of the radiolytic molecules in case 2, which increased the computational burden during the search for potential reactants of each molecule. As compared to Geant4-DNA using a single CPU core, the speed-up factor achieved by our package was 171.1 for case 1 and 197.2 for case 2. The occupancy of GPU multiprocessors in our simulation was estimated to be 100% using the occupancy calculator provided by Nvidia. The speed-up factor per multiprocessor was 11.4 and 13.1 for the two cases, respectively. We would like to point out that this presented acceleration factor is probably unfair to Geant4-DNA, because Geant4-DNA also recorded other variables for each molecule, such as the track ID and parent ID to be able to track the path of the molecule during the simulation if needed.

4. Discussions and Conclusions

The accurate simulation of water radiolysis is an essential step to understand the radiobiological mechanisms and test the hypotheses in related problems. MC simulation of the chemical stage is highly time-consuming because of the large number of simulation time steps necessary to cover this stage as compared with physical and prechemical water radiolysis stages. The large number of molecules and high code complexity to track interactions among them also prolong the computation. We reported our initial development on gMicroMC, a GPU-based fast microscopic MC simulation package for water radiolysis. The work presented in this paper focused on accelerating MC simulation of the chemical stage on GPU. We presented the details of our GPU-friendly parallelization strategy specifically designed for this highly correlated many-body problem. The yields of the radiolytic species for a 750 keV electron and a 5 MeV proton were used to validate the accuracy of our simulation. The relative differences between our simulation and the Geant4-DNA simulation were on average 5.31% and 4.43% for the two cases. Executed on an Nvidia GTX Titan black GPU card, our gMicroMC achieved a speed-up factor of 171.1 and 197.2 over Geant4-DNA that was executed on an Intel i7-5500U CPU processor. This has demonstrated that the MC simulation of the chemical stage in water radiolysis can be substantially accelerated using the powerful parallelization capability of GPU when a large number of radiolytic molecules are involved.

Parallelizing the simulation of the chemical stage on GPU is more challenging than other typical GPU-based MC particle transport simulation tasks, such as radiotherapy dose calculation and the simulation of fluids or contacting particle systems. In the MC dose calculation, transport simulation is conducted independently for each radiation particle,

including secondary particles. Hence parallelization on GPU is relatively straightforward, although some GPU-friendly strategies are favored to alleviate the GPU thread divergence issue due to the highly diverse physical interactions between different particles (Jia *et al.*, 2014). In the simulation of fluids or contacting particle systems, a particle receives an interaction force from its neighbors, making the task a many-body simulation problem. Nevertheless, when using small time steps, the total interaction force and the velocity change for each particle may still be calculated independently, based on the status of the particles at the previous time step (Green, 2010). In contrast, for the chemical stage in water radiolysis, the simulation of each molecule depends on that of other molecules because of possible mutual chemical reactions among the radiolytic molecules; this process challenges parallelization. In addition, the chemical stage simulation involves a continually changing number of radiolytic species and a huge number of grid cells for the grid data, requiring a careful memory arrangement on GPU. A GPU-friendly parallelization strategy was specifically designed to address these challenges.

The presented study was only about our development on the simulation of the chemical stage in water radiolysis. To complete the gMicroMC package, we are currently conducting developments on the simulations for the physical stage and the pre-chemical stage. It will be also our future study to add other biologically related functions into our gMicroMC package, including the modeling of DNA and calculations of DNA damage, such as single-strand break and double-strand break (Schuermann, 2012). We are open to collaborations to refine our chemistry models and test this package. Upon completion, we will make the gMicroMC package an open source to our community.

Acknowledgments

This project is supported in part by NIH grant (P20CA183639-01A1). The authors would like to thank Dr. Mathieu Karamitros for his helpful discussions on the use of Geant4-DNA and Dr. Damiana Chiavolini for editing this manuscript.

References

- Alper T. The modification of damage caused by primary ionization of biological targets. *Radiation Research*. 1956; 5:573–86. [PubMed: 13379615]
- Alper T, Bryant P. Reduction in oxygen enhancement ratio with increase in LET: Tests of two hypotheses International. *Journal of Radiation Biology and Related Studies in Physics, Chemistry and Medicine*. 1974; 26:203–18.
- Baverstock K, Burns W. Oxygen as a product of water radiolysis in high-LET tracks. II. Radiobiological implications. *Radiation research*. 1981; 86:20–33. [PubMed: 7012902]
- Bell N, Hoberock J. Thrust: A productivity-oriented library for CUDA. GPU computing gems Jade edition. 2011; 2:359–71.
- Bernal M, Bordage MC, Brown J, Davidková M, Delage E, El Bitar Z, Enger S, Francis Z, Guatelli S, Ivanchenko V. Track structure modeling in liquid water: A review of the Geant4-DNA very low energy extension of the Geant4 Monte Carlo simulation toolkit. *Physica Medica*. 2015; 31:861–74. [PubMed: 26653251]
- Dematte L. Smoldyn on graphics processing units: massively parallel Brownian dynamics simulations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*. 2012; 9:655–67.
- Friedland W, Dingfelder M, Kunderát P, Jacob P. Track structures, DNA targets and radiation effects in the biophysical Monte Carlo simulation code PARTRAC. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*. 2011; 711:28–40. [PubMed: 21281649]

- Gladkov, DV., Alberts, S., D'Souza, RM., Andrews, S. Proceedings of the 19th High Performance Computing Symposia. Vol. Series. Society for Computer Simulation International; 2011. p. 151-8.
- Green S. Particle simulation using cuda. NVIDIA whitepaper. 2010; 6:121–8.
- Halliwell B, Aruoma OI. DNA damage by oxygen - derived species Its mechanism and measurement in mammalian systems. FEBS letters. 1991; 281:9–19. [PubMed: 1849843]
- Hamm R, Turner J, Stabin M. Monte Carlo simulation of diffusion and reaction in water radiolysis—a study of reactant jump through and jump distances. Radiation and environmental biophysics. 1998; 36:229–34. [PubMed: 9523338]
- Hirayama R, Ito A, Tomita M, Tsukada T, Yatagai F, Noguchi M, Matsumoto Y, Kase Y, Ando K, Okayasu R. Contributions of direct and indirect actions in cell killing by high-LET radiations. Radiation research. 2009; 171:212–8. [PubMed: 19267547]
- Hissoiny S, Ozell B, Bouchard H, Després P. GPUMCD: A new GPU-oriented Monte Carlo dose calculation platform. Medical physics. 2011; 38:754–64. [PubMed: 21452713]
- Incerti S, Baldacchino G, Bernal M, Capra R, Champion C, Francis Z, Guèye P, Mantero A, Mascialino B, Moretto P. The Geant4-DNA project. International Journal of Modeling, Simulation, and Scientific Computing. 2010; 1:157–78.
- 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. Phys Med Biol. 2010; 55:3077. [PubMed: 20463376]
- Jia X, Schumann J, Paganetti H, Jiang SB. GPU-based fast Monte Carlo dose calculation for proton therapy. Phys Med Biol. 2012; 57:7783–97. [PubMed: 23128424]
- Jia X, Ziegenhein P, Jiang SB. GPU-based high-performance computing for radiation therapy. Phys Med Biol. 2014; 59:R151–82. [PubMed: 24486639]
- Kalantzis G, Emfietzoglou D, Hadjidoukas P. A unified spatio-temporal parallelization framework for accelerated Monte Carlo radiobiological modeling of electron tracks and subsequent radiation chemistry. Computer Physics Communications. 2012; 183:1683–95.
- Karamitros M, Luan S, Bernal M, Allison J, Baldacchino G, Davidkova M, Francis Z, Friedland W, Ivantchenko V, Ivantchenko A. Diffusion-controlled reactions modeling in Geant4-DNA. Journal of Computational Physics. 2014; 274:841–82.
- Karamitros M, Mantero A, Incerti S, Friedland W, Baldacchino G, Barberet P, Bernal M, Capra R, Champion C, Bitar Z. Modeling radiation chemistry in the Geant4 toolkit. Progress in nuclear science and technology. 2011; 2:503–8.
- Kreipl MS, Friedland W, Paretzke HG. Time- and space-resolved Monte Carlo study of water radiolysis for photon, electron and ion irradiation. Radiation and environmental biophysics. 2009; 48:11–20. [PubMed: 18949480]
- Michalik V, Begusova M, Bigildeev E. Computer-aided stochastic modeling of the radiolysis of liquid water. Radiation research. 1998; 149:224–36. [PubMed: 9496885]
- Naqvi KR, Mork K, Waldenstrom S. Diffusion-controlled reaction kinetics. Equivalence of the particle pair approach of Noyes and the concentration gradient approach of Collins and Kimball. The Journal of Physical Chemistry. 1980; 84:1315–9.
- Neary G. Chromosome aberrations and the theory of RBE: 1. General considerations. International Journal of Radiation Biology and Related Studies in Physics, Chemistry and Medicine. 1965; 9:477–502.
- Nikjoo H, Uehara S, Emfietzoglou D, Cucinotta F. Track-structure codes in radiation research. Radiation Measurements. 2006; 41:1052–74.
- O'Neill P, Wardman P. Radiation chemistry comes before radiation biology. International journal of radiation biology. 2009; 85:9–25. [PubMed: 19205982]
- Plante I. A Monte-Carlo step-by-step simulation code of the non-homogeneous chemistry of the radiolysis of water and aqueous solutions—Part II: calculation of radiolytic yields under different conditions of LET, pH, and temperature. Radiation and environmental biophysics. 2011a; 50:405–15. [PubMed: 21594646]
- Plante I. A Monte-Carlo step-by-step simulation code of the non-homogeneous chemistry of the radiolysis of water and aqueous solutions. Part I: theoretical framework and implementation. Radiation and environmental biophysics. 2011b; 50:389–403. [PubMed: 21562854]

- Schuemann J. SU - E - T - 475: Nano - Dosimetric Track Structure Scoring including Biological Modeling with TOPAS - NBio. Medical Physics. 2012; 39:3814.
- Tachibana H. Accelerated event-by-event Monte Carlo microdosimetric calculations of electrons and protons tracks on a multi-core CPU and a CUDA-enabled GPU. Computer methods and programs in biomedicine. 2014; 113:116–25. [PubMed: 24113420]
- Uehara S, Nikjoo H. Monte Carlo simulation of water radiolysis for low-energy charged particles. Journal of radiation research. 2006; 47:69–81. [PubMed: 16571920]

Figure 1.
Flowchart of our GPU-based MC simulation of the chemical stage.

Figure 2.

A 2D example to illustrate the basic idea of the grid data. (a) The 2D VOI containing 11 molecules (denoted by solid circles) was divided into a 4×4 uniform grid. The numbers in the upper left corner of each grid cell denote the cell IDs, and the ones next to molecules denote the molecule IDs. The shaded nine cells were the cells in which the reactant search would be conducted for molecule 7. (b) The grid data built for this example, including the sorted molecule array, the non-empty-cell array, the start-position array, and the look up table.

Figure 3.

Live radiolytic molecules at different time steps during the chemical stage for the two testing cases. Rows a-b represent case 1 (a 750 keV incident electron) and case 2 (a 5 MeV incident proton), respectively. The initial direction of the incident particle was z direction. Column 1-3 correspond to three time points at 1 ps, 10^5 ps and 10^6 ps. To better demonstrate the diffusion of the molecules, only a subsection of the radiation track that we simulated is shown here.

Figure 4.

Time-dependent yields of species produced in water by a 750 keV electron. The yields of species $\cdot\text{OH}$, e_{aq}^- , $\text{H}\cdot$ and H_2O_2 are shown in the left panel, and the yields of species OH^- , H_2 and H_3O^+ are shown in the right panel. The PARTRAC results were from (Kreipl et al., 2009). Only the yields of $\cdot\cdot\text{OH}$, e_{aq}^- , H_2O_2 and H_2 were reported in this reference.

Figure 5.

Time-dependent yields of species produced in water by a 5 MeV proton. The yields of species $\cdot\text{OH}$, e_{aq}^- , $\text{H}\cdot$ and H_2O_2 are shown in the left panel, and the yields of species OH^- , H_2 and H_3O^+ are shown in the right panel. The PARTRAC results were from (Kreipl *et al.*, 2009). Only the yields of $\cdot\text{OH}$, e_{aq}^- , H_2O_2 and H_2 were reported in this reference.

Table 1

Fixed step sizes of the time steps used in our simulation.

Time interval (s)	Fixed step sizes Δt (ps)
$10^{-12} \sim 10^{-11}$	0.1
$10^{-11} \sim 10^{-10}$	1
$10^{-10} \sim 10^{-9}$	3
$10^{-9} \sim 10^{-8}$	10
$10^{-8} \sim 10^{-6}$	100

Table 2
MC simulation workflow for the chemical stage in water radiolysis

The start time t was set to 1 ps. Conduct the following procedures at each time step until the chemical stage ends (i.e. $t = 1 \mu\text{s}$).	
1	Based on fixed step size listed in Table 1, compute reaction radius using Eq. (2). Search for potential reactant pairs;
2	Determine the actual step size Δt : <ul style="list-style-type: none">• If there is any potential reactant pair satisfying $d \leq R$, simulate the reaction. Set $\Delta t = 0$ and go to step 6);• If no reaction occurs, calculate Δt_D for each pair using Eq. (4) and $\Delta t = \min(\Delta t_D)$.
3	Diffuse the molecules using the probability in Eq. (1);
4	Search for potential reactant pairs based on the molecule positions after diffusion;
5	Determine whether to simulate a reaction for each pair: <ul style="list-style-type: none">• For a potential pair satisfying $d \leq R$, simulate the reaction;• For a potential pair satisfying $d > R$, use Brownian bridge method to determine whether a crossing event occurs. If yes, simulate the reaction.
6	Set $t = t + \Delta t$; go to step 1).

Table 3
Radiolytic molecule species types and their diffusion coefficients

Species	Diffusion coefficient D ($10^{-9}m^2s^{-1}$)
e_{aq}^-	4.9
$\cdot OH$	2.8
$H \cdot$	7.0
H_3O^+	9.0
H_2	4.8
OH^-	5.0
H_2O_2	2.3

Table 4
Default reaction types and reaction rate constants

Reaction*	Reaction rate constant ($10^{10}\text{M}^{-1}\text{s}^{-1}$)
$\text{H} \cdot + \text{e}_{\text{aq}}^- + (\text{H}_2\text{O}) \rightarrow \text{OH}^- + \text{H}_2$	2.65
$\text{H} \cdot + \cdot \text{OH} \rightarrow (\text{H}_2\text{O})$	1.44
$\text{H} \cdot + \text{H} \cdot \rightarrow \text{H}_2$	1.20
$\text{H}_2 + \cdot \text{OH} \rightarrow \text{H} \cdot + (\text{H}_2\text{O})$	0.00417
$\text{H}_2\text{O}_2 + \text{e}_{\text{aq}}^- \rightarrow \text{OH}^- + \cdot \text{OH}$	1.41
$\text{H}_3\text{O}^+ + \text{e}_{\text{aq}}^- \rightarrow \text{H} \cdot + (\text{H}_2\text{O})$	2.11
$\text{H}_3\text{O}^+ + \text{OH}^- \rightarrow (2\text{H}_2\text{O})$	14.3
$\cdot \text{OH} + \text{e}_{\text{aq}}^- \rightarrow \text{OH}^-$	2.95
$\cdot \text{OH} + \cdot \text{OH} \rightarrow \text{H}_2\text{O}_2$	0.44
$\text{e}_{\text{aq}}^- + \text{e}_{\text{aq}}^- + (2\text{H}_2\text{O}) \rightarrow 2\text{OH}^- + \text{H}_2$	0.50

* H_2O molecules in the chemical equations were ignored in our simulation assuming they were everywhere.

Table 5

Yield values (molecules/100 eV) of radiolytic species present at 1 μ s for 750 keV electron and 5 MeV proton.

	$\frac{-}{e_{aq}}$	$\cdot OH$	$H \cdot$	H_3O^+	H_2	OH^-	H_2O_2		
750 keV electron	Geant4-DNA	gMicroMC Difference (%)	2.51	3.03	0.66	3.00	0.49	0.39	0.45
			2.51	3.07	0.72	3.06	0.53	0.36	0.41
			0.00	1.32	9.09	2.00	8.16	7.69	8.89
5MeV proton	Geant4-DNA	gMicroMC Difference (%)	1.38	1.97	0.73	2.05	0.63	0.66	0.70
			1.40	1.97	0.80	2.15	0.60	0.70	0.67
			1.45	0.00	9.59	4.88	4.76	6.06	4.29

Table 6

Efficiency test results, including the amount of radiolytic molecules at the beginning of the chemical stage in the two cases, denoted as $N(t)$, $t = 1\text{ps}$; total simulation time taken by Geant4-DNA on CPU and our gMicroMC on GPU; the speed-up factor achieved by our package compared to the Geant4-DNA.

	$N(t) \ t = 1\text{ps}$	Simulation time (s)		Speed-up factor
		Geant4-DNA	gMicroMC	
Case 1	101829	102865.4	599.2	171.1
Case 2	56122	96446.5	489.0	197.2