Fast Calculation of Mont Carlo Ion Transport Code

Meiling Qi, Qiong Yang*, Xiaolong Chen, Jizheng Duan and Lei Yang
Institute of Modern Physics, Chinese Academy of Sciences, 730000, Lanzhou, China
Email: yangqiong@impcas.ac.cn

Abstract. In this work, a Monte Carlo code has been developed for quick simulation of carbon ion bombardment in materials. This program realizes the calculation of the displacement damage and ionization of particle (<10MeV/amu) in materials. In addition, high energy carbon (<500MeV/amu) transport calculation is developed to simulate the ionization dose in water. The GPU parallel algorithm is adopted to speed up the program. Subsequently, the vacancies and ionization of target materials induced by various energy carbon ion were simulated and compared with the SRIM2013. Dose depth distributions of different energy carbon in water were simulated and compared with GSI experimental data. Results of simulation are consonant well with SRIM2013 and experimental data. It achieves significant speed-up ratio in different GPU devices.

1. Introduction
GPU (Graphics Processing Unit) is widely used in high performance computing for its superiority of good parallelism, low energy consumption ratio and cost [1]. TRIM (Transport of Ions in Matter) is a Monte Carlo (MC) program that follows the ion into the target, tracks detailed trajectories of incident ions colliding with target atom, which simulates the transport of ion in various matters. It is extensively applied in the damage and ionization simulation of materials irradiated by medium and low energy ions.

Carbon ion is recognized as the most suitable heavy ion in radiotherapy for its biological benefits such as enhanced Relative Biological Effectiveness (RBE) in the Bragg peak region, low lateral scattering and Oxygen Enhancement Ratio (OER) and large proportion of DNA double-strand break. The carbon ion beam with energy below 500 MeV/amu is appropriate for therapy owing to its proper dose Bragg peak range in human tissue. It is different from low energy carbon ion transport. The dose distribution caused by hundreds of MeV carbon ions is largely affected by the fragments of nuclear reaction with human tissue elements [2]. Therefore, the generation and transport of nuclear reaction fragments between carbon ions and human elements must be considered in carbon therapy simulations. Based on the TRIM theory, GPU Accelerated Carbon ion Transport (GACT) is developed to realize preliminary rapid assessments of the dose and displacement damage of carbon ions. In addition, high energy carbon ion transport cross section is added to the program to calculate the dose of carbon ions.

2. Physics and Methods
Monte Carlo (MC) method has the powerful ability to deal with complex geometry and materials, It is a powerful tool for simulation and design in the field of nuclear physics and nuclear engineering. Continuous Slowing Down Approximation (CSDA) method [3] is usually used to treat the continuous collision process of ions in matter. MC method and CSDA method are used as basic physical algorithm in this code. Each particle’s track is spited into a number of sub steps. In each step,
ionization energy loss at each track is assumed to be equal to the plus of stopping power and track length.

2.1. Electronic and Nuclear Stopping
The energy loss of ions transport in the medium comes from two processes: electronic stopping and nuclear stopping. Firstly, the electronic stopping cross-sections of proton are obtained using the Lindhard local-density approximation [3]. For helium ion, the stopping cross-sections are obtained by multiply the effective helium charge by the corresponding proton data. The heavy ions cross-section data are obtained by scaling proton stopping powers according to the Brandt-Kitagawa theory [2]. Bethe-Bloch formula [4] is adopted to get electronic stopping data when the ion energy is above 20MeV/amu.

Nuclear stopping comes from the elastic scattering process of ion and target nuclei. In the process of particle nucleon scattering, the target nucleus accept part of the energy of ions, and the direction of ion motion will change after collision. The target nucleus with enough kinetic energy, will leave the original position of the lattice, causing dislocation damage. Universal Interatomic Potential [3] is adopted to treat the nucleon scattering in the code.

2.2. Kinchin-Pease Model
In quick GACT, the modified Kinchin-Pease theory is used to calculate target material displacements caused by incident ions and cascade target nucleus. Norgett, Robinson, and Torrens [4] proposed a formula for calculating the number of displacement primary knock-on atom (PKA) which is often referred to as the NRT model. The number of displacements $N_v$ is derived from the NTR equation:

$$
N_v = \begin{cases} 
0 & E_v < E_d \\
1 & E_d \leq E_v < 2.5E_d \\
0.8\frac{E_v}{2E_d} & E_v \geq 2.5E_d
\end{cases}
$$

(1)

Where, $E_v$ is the damage energy and $E_d$ is the threshold displacement energy.

2.3. Charged Fragments Cross Section Build-Up
The dose distribution caused by the hundreds of MeV carbon ions is largely affected by the fragments of nuclear reaction with human tissue elements [2]. Therefore, a database of fragments yield for those elements bombed by carbon ion is created. The data recorded information about the energy of incident carbon ion and the corresponding fragment types, energy and angular distributions. Once a carbon ion collides with an atom during the transport process, secondary particles will be produced. For each type of secondary particles, the code performed random process to obtain the yield, energy and scattering angle according to the database. In addition, all electrically neutral particles generated in nuclear interactions is ignored for its very low contribution to dose, and it is a clinically acceptable approximation for treatment planning purposes.

3. GPU Parallel Algorithm
GPU executes programs in single-instruction-multiple-data (SIMD) mode, which is suitable for data parallelism. The physical process in this work is naturally parallelism, because the particle transport process is relatively independent and the random process is simple. The structure of the GACT is shown in figure 1, which is founded on CUDA architecture of NVIDIA. CPU controls the program, GPU completes all MC computing tasks. GACT uses equal division, since the calculation scale of MC transport is large and each computing task is independent. That is, the calculation task is divided into $np / nt$, where $np$ is the total number of jobs, $nt$ is the number of threads on GPU. The job number is increased by $ni = ni + blockSize \times gridSize$, where $ni$ is the job number, blockSize is the
size of block and gridSize is the size of the grid.

![Diagram of GACT parallel algorithm]

**Figure 1.** GACT parallel algorithm.

4. Simulation Parameters and Results

4.1. Dissociation Number and Ionization Energy Loss Caused by Carbon Ions

Iron is chosen as the calibration material for it is the one of the common nuclear materials elements. For the verification and acceleration efficiency comparison, we used SRIM2013 to simulate the damage and ionization distribution in iron, and execution times were also recorded. The ionization and vacancy distribution caused by carbon ions in different energy are illustrated in figure 2 and figure 3. The identical damage estimation model is used in GACT and SRIM2013 simulations. We can see that the results of GACT are consistent with SRIM2013.
Figure 2. The ionization distribution along iron target depth caused by carbon with different energies.

Figure 3. Vacancy distribution along iron target depth caused by carbon with different energies.

4.2. Bragg Peak of Carbon Ions in Water

We compared the axial dose distribution of carbon particles with different energies in water with the GSI experimental data [5, 6]. In this simulation, the water target was set as a cylinder, the radius was 10cm and the length 40cm. The energy of carbon ion was set to 135 MeV/amu, 195 MeV/amu, 270 MeV/amu and 330 MeV/amu. Four cases of different carbon energy were set to perform $3 \times 10^7$ particle simulations to get sufficient statistics. Figure 4 shows the comparisons between the simulation results of GACT and the experimental results. This indicates the axial distribution curve of dose deposition caused by different energy carbon particles shows the Bragg peak distribution trend obviously. The GPU code simulated position and height of Bragg peak are consistent with the experimental data.
Figure 4. Comparisons of calculated and measured $^{12}\text{C}$ depth-dose profiles in water (135 MeV/amu, 195 MeV/amu, 270MeV/amu and 330MeV/amu)

4.3. Speedup Test of GACT
NVIDIA TITAN V, Tesla P100, Titan RTX and RTX2080 Ti were selected as the GACT test platforms. SRIM2013 run on ThinkPad-X1 laptop platform with standard voltage CPU i7 8750H. Table 1 shows good speedup results between SRIM2013 and GACT in the calculation of $10^5$ particles, and the speedup ratio can reach 3000 to 10000 times.

Table 1. Comparison of time consumption of simulate $10^5$ particles between SRIM2013 and GACT.

| Calculate time(second) | 12.5KeV/amu | 41.6KeV/amu | 83.3KeV/amu | 0.83MeV/amu |
|------------------------|-------------|-------------|-------------|-------------|
| GACT (NVIDIA TITAN V)  | 0.846       | 0.857       | 0.851       | 0.876       |
| GACT (NVIDIA Tesla P100)| 0.824       | 0.826       | 0.851       | 0.917       |
| GACT (NVIDIA TITAN RTX)| 1.23        | 1.24        | 1.31        | 1.59        |
| GACT (GeForce RTX2080 Ti)| 1.34        | 1.47        | 1.53        | 1.57        |
| SRIM2013 (Intel Core i7 8750h)| 4322        | 5649        | 6547        | 10088       |
| Time Gain Factor (*$10^3$) | 3.2~8.7     | 3.8~11.8    | 4.3~7.7     | 6.3~11.5    |

Four cases of high energy carbon were simulated to perform $3\times10^7$ particle simulations in the same GPU platforms. By contrast, FLUKA was used to perform the same calculation under the same calculation model and carbon particle energy. FLUKA run on the CPU computing cluster, the CPU was Intel Xeon E5-2630 v4, using 40 cores in parallel.
### Table 2. Time consumption of FLUKA and GACT in different platforms.

| Simulation time (Minute) | 135 MeV/amu | 195 MeV/amu | 270 MeV/amu | 330 MeV/amu |
|--------------------------|-------------|-------------|-------------|-------------|
| GACT (NVIDIA TITAN V)    | 0.58        | 1.43        | 2.21        | 3.41        |
| GACT (NVIDIA Tesla P100) | 0.56        | 1.39        | 2.18        | 3.69        |
| GACT (NVIDIA TITAN RTX)  | 1.65        | 4.16        | 6.29        | 9.8         |
| GACT (GeForce RTX2080 Ti)| 1.79        | 4.33        | 6.81        | 10.1        |
| FLUKA (Intel(R) Xeon E5-2640) | 195.2 | 411.3 | 448.4 | 612.2 |
| Time gain factor         | 109–336     | 97–295      | 66–206      | 61–179      |

We can see from table 2: (1) With the increase of carbon ion energy, the calculation time is also increasing, this is because at high energy, the number of secondary fragments increases and the range of particles in the material becomes deeper, which results in the increase of calculation. (2) The simulation time is affected by the performance of GPU. The higher the computing power and the larger the memory, the simulation time is shorter.

### 5. Conclusion and Discussion

Based on CUDA, the carbon ion transport code GACT is developed. The result of GACT is in good agreement with SRIM2013 and experimental data. The program can be used for rapid damage analysis of materials for incident ion with energy ranged from eV to several MeV. Moreover, the calculation of the dose of carbon ions in water has been initially achieved, and the program will continue to be improved in the future.

### Acknowledgments

The authors acknowledge the Supercomputing Center of Institute of Modern Physics, Chinese Academy of Sciences for providing the GPU test platform, and the team members for their guidance on the TRIM physics algorithm. This work is supported by Strategic Priority Research Program of the Chinese Academy of Sciences, grant XDA21010202.

### References

[1] CURAND LIBRARY Programming Guide 2019 PG-05328-050_v Release Version
[2] Monika Puchalska, et al. 2018 Benchmarking of PHITS for carbon ion therapy International Journal of Particle Therapy 4(3): 48-55.5
[3] Ziegler J F, Biersack J P, Littmark U 1985 The Stopping and Range of Ions In Solids Pergamon Press, New York
[4] Groom D E, et al. 2000 Passage of particles through matter Eur. Phys. J. C 15: 163–173
[5] H. Iwase, K, et al. 2007 Experimental and theoretical study of the neutron dose produced by carbon ion therapy beams Radiation Protection Dosimetry 126: 615–618
[6] Lembit Sihver, Deter Schardt, Tatsuki Kanai. et al. 1998 Depth-dose distributions of high-energy carbon, oxygen and neon beams in water Jpn. J. Med. Phys 18(1): 1-21