ARTICLE IN PRESS

Computer Physics Communications ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

A graphics processor-based intranuclear cascade and evaporation simulation

H. Wan Chan Tseung*, C. Beltran

Division of Medical Physics, Department of Radiation Oncology, Mayo Clinic, Rochester MN 55905, United States

ARTICLE INFO

Article history:
Received 7 October 2013
Received in revised form
10 March 2014
Accepted 8 April 2014
Available online xxxx

Keywords:
Monte Carlo
Proton transport
Nuclear cascade
Evaporation
GPU
CUDA

ABSTRACT

Monte Carlo simulations of the transport of protons in human tissue have been deployed on graphics processing units (GPUs) with impressive results. To provide a more complete treatment of non-elastic nuclear interactions in these simulations, we developed a fast intranuclear cascade-evaporation simulation for the GPU. This can be used to model non-elastic proton collisions on any therapeutically relevant nuclei at incident energies between 20 and 250 MeV. Predictions are in good agreement with Geant4.9.6p2. It takes approximately 2 s to calculate 1×10^6 200 MeV proton–^{16}O interactions on a NVIDIA GTX680 GPU. A speed-up factor of ∼20 relative to one Intel i7-3820 core processor thread was achieved.

1. Introduction

Monte Carlo (MC) techniques are increasingly being used in proton therapy to simulate and validate treatment plans [1,2]. MC calculations are more accurate than pencil-beam evaluations, since they take into account detailed microscopic processes and have a better handling of material inhomogeneities in patients. However, computational times associated with MC simulations can be prohibitive. For example, calculating the dose within 2% statistical error in a 150 cm^3 volume requires around 1 h on a modern 100-node cluster.[1] At present, the use of MC is therefore either restricted to institutions that have access to significant computing resources, or reserved for a handful of cases requiring detailed investigation and enhanced accuracy.

Recently, there have been several attempts to simulate both charged particle and photon transport on graphics processing units (GPUs) [5–8]. In particular, proton transport MCs for particle therapy applications have been successfully implemented. Using GPUs, MC calculations of treatment plans involving 1×10^7 proton trajectories have been completed in under 30 s. It is therefore clear that GPUs can be of tremendous benefit to proton therapy, both for research and in clinical applications (e.g. plan validation and optimization).

However, the current generation of GPU proton MCs make a number of simplifying assumptions concerning non-elastic nuclear interactions. Kohno et al. [5] do not directly simulate nuclear processes, but 'include' them by using measured proton depth dose distributions in water. Jia et al. [8] follow Fippel and Soukup [9] by using a crude model of nuclear interactions on the oxygen nucleus. They demonstrate that the lack of an in-depth model of nuclear processes does not significantly impact dose predictions in low-density tissue and bone. It is nonetheless conceivable that unsatisfactory results could be obtained in certain situations, e.g. propagation through high-Z materials such as metallic implants. Detailed investigations (e.g. studies of secondary particles and residual nuclei) are also not possible with the model used in Refs. [8,9].

To better include non-elastic nuclear interactions, one can envisage a hybrid proton transport program, in which ionization energy loss, straggling and multiple scattering are simulated on the GPU, while nuclear events are handled by existing software packages on the CPU. However, in this configuration the running time is expected to be completely dominated by nuclear calculations. For 1×10^7 protons in the therapeutic energy range (70–250 MeV), we estimate the number of nuclear events that need to be simulated to be of the order of 10^6. On a i7-3820 3.6 GHz processor it can take from 200 to 900 s to simulate 1×10^6 200 MeV proton–^{16}O non-elastic interactions with Geant4.9.6p2, depending on the chosen nuclear interaction model.

Our primary aim is to improve the treatment of nuclear processes in a GPU proton transport MC without considerably

* Corresponding author.
E-mail address: wanchantseung.hok@mayo.edu (H. Wan Chan Tseung).

[1] This simulation was carried out with TOPAS [3], a proton therapy simulation package that facilitates the use of Geant4 [4].

http://dx.doi.org/10.1016/j.cpc.2014.04.007
0010-4655/

extending the net calculation time. Thus, we developed a rudimentary but fast GPU-based MC simulation of nuclear interactions, which is able to predict secondary particle properties following non-elastic events for incident proton energies below 250 MeV. To our knowledge, such simulations have not been previously reported. We show that on a NVIDIA GTX680 card, 1×10^6 200 MeV proton–^{16}O nuclear events can be computed in 2 s.

This paper is organized as follows. In Section 2, we describe our approach, which is a conventional Bertini-type intranuclear cascade (INC) model including nuclear evaporation. We then give details of its implementation on the GPU in Section 3. Our results are described in Section 4. Predictions from our code are verified with Geant4.9.6p2; simulated data for proton interactions with ^{16}O and ^{40}Ca are shown, and calculation times are compared. We summarize in Section 5.

2. Non-elastic interaction model

2.1. Overview

A non-elastic nucleon–nucleus interaction can be assumed to take place in two steps: a fast INC stage that leaves the nucleus in an excited state, followed by an evaporation stage. The INC phase has been modeled extensively since the late 1950s using MC techniques [10–14]. In the Bertini approach [11], the incident particle and nucleon–nucleon collision products are tracked, assuming straight line trajectories and taking into account medium effects, until they exit the nucleus or their energies lie below a cutoff. The simulated nucleon–nucleon collisions are not ordered in time. In the evaporation phase, de-excitation of the nucleus occurs after energy equilibration, first through the emission of low-energy particles (nucleons and possibly heavier fragments), then photons. The probability of particle emission can be calculated using statistical methods [15,16].

The INC-evaporation model applies to non-elastic interactions in the therapeutic energy range down to a few tens of MeV, although its validity becomes questionable at low energy. It contains no adjustable parameters, i.e. its predictions do not need to be normalized. Our model inputs, assumptions and further details on the two phases are given below.

2.2. Nuclear model

The nucleus is treated as a two-component fermion gas mixture of protons and neutrons. The nucleon density $\rho(r)$ follows a Fermi distribution:

$$\rho(r) \propto (1 + \mathrm{e}^{\frac{r-c}{\delta}})^{-1} \tag{1}$$

where $c = 1.07A^{\frac{1}{3}}$ (A is the mass number), and $\delta = 0.545$. For simplicity, $\rho(r)$ is approximated by a step function consisting of 15 shells of constant densities and equal thickness. The density of each shell is found by integrating $\rho(r)$. Following Chen et al. [12], we take the radius of a nucleus to be $c + 2.5$ fm. The potential energy V of a nucleon is the sum of its Fermi energy E_F and the binding energy E_B, which is calculated using the semi-empirical mass formula. Target nucleon momenta are sampled from a quadratic distribution.

2.3. Intranuclear cascade

As mentioned above, the incident nucleon and collision products are followed in the nucleus. The sequence of calculations in our cascade simulation roughly follows that of Metropolis et al. (Fig. 3 in Ref. [10]). Below 250 MeV, only nucleon–nucleon elastic collisions need to be considered. Total and differential elastic nucleon–nucleon cross-sections are calculated using the parameterizations of Cugnon et al. [17]. Collision kinematics are fully relativistic. Pauli blocking is enforced by requiring that the kinetic energies of all collision products exceed their Fermi energies. The cutoff energy is taken to be $E_F + E_B$ for neutrons and $E_F + E_B + E_C$ for protons, where E_C is the Coulomb barrier. Refraction and reflection of nucleons at shell boundaries are taken into consideration using the prescription of Chen et al. [12]. At the end of the cascade, the excitation energy of the residual nucleus is calculated through energy conservation, as in Ref. [10].

2.4. Nucleon evaporation

For the evaporation phase, we adopt the generalized evaporation model (GEM) [18]. Although the GEM can handle nuclide emission up to Mg, only He and lighter particles are considered in this work. An isotropic angular distribution is assumed in the rest frame of the nucleus for all evaporated particles. Other post-cascade processes such as photon evaporation, pre-equilibrium emission of particles, fission and fermi break-up are not currently included in our model.

3. GPU implementation

In this section, our implementation of the above model on a GPU[2] using the CUDA framework [20] is described.

3.1. Software organization and memory allocation

Three kernels are used for the following tasks: (1) initialization of random number sequences for generation within individual threads using the CURAND library [21], (2) the INC simulation, and (3) nuclear evaporation calculations. Kernel 2 processes one full cascade and kernel 3 computes one nuclear de-excitation per GPU thread.

As for memory allocation, GPU constant memory is used to store simulation-wide physics constants. Pre-calculated total nucleon–nucleon cross-section tables and other read-only input data arrays are stored as 1-D textures. Global memory is used to store information on incident particles, excited nuclei, and secondary particles that escape the nucleus. Collision products are stored in per-thread local memory before being simulated. Shared memory is minimally used in the present work. Floating point precision is adopted in all calculations.

Compile-time conditions are inserted to isolate CUDA C extensions, so that the same program can be made to run on the CPU. This greatly facilitates debugging and running time comparisons.

3.2. Intranuclear cascade kernel

The INC model described in Section 2.3 was implemented as one GPU kernel. The inputs to this kernel are: a structure of arrays (SoA) describing the incident protons (position, momentum, and energy) and the initial CURAND states for all threads. A SoA is used to ensure coalesced global memory reads. The outputs are two SoAs containing information on exiting particles and the excited nuclei remaining after the INC.

The limiting factors affecting run-time performance were: (1) thread divergence caused by the intrinsic stochastic nature of the simulation and conditional instructions, (2) register spill and local memory overhead, due to the large kernel size and the use of local memory arrays to store secondary nucleons that need to be propagated in the INC.

[2] We assume the reader to be acquainted with GPU terminology; if not, one can refer to the abundant resources available on the NVIDIA developer website [19].

Please cite this article in press as: H. Wan Chan Tseung, C. Beltran, A graphics processor-based intranuclear cascade and evaporation simulation, Computer Physics Communications (2014), http://dx.doi.org/10.1016/j.cpc.2014.04.007