

Development of a GPU-enhanced time-dependent Monte Carlo neutron transport version of McCARD

Woo Kyoung Ko, Seong Jeong Jeong, Young In Kim, and Hyung Jin Shim* 

Nuclear Engineering Department, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea

Received: 4 August 2024 / Received in final form: 5 November 2024 / Accepted: 15 November 2024

Abstract. To enhance the performance of the TDMC neutron transport analysis, a new version of McCARD utilizing GPUs, named McCARD/G, is under development. This version introduces an MC neutron tracking module that employs an event-based algorithm and features a generalized lattice geometry treatment module. Additionally, significant adaptations of other major modules have been made to optimize for the GPU architecture. The capabilities of McCARD/G have been verified through the C5G7-TD benchmark, which shows good agreements with the reference results. Furthermore, McCARD/G's efficacy is also demonstrated via an experimental benchmark conducted at the Kyoto University Critical Assembly.

1 Introduction

The transient analysis of nuclear reactors is an ultimate development direction for Monte Carlo (MC) neutron transport analysis codes. Since the introduction of the time-dependent MC (TDMC) neutron tracking method [1], which updates the collision time of neutron according to its speed and flight length, significant advancements in TDMC methodologies have been made recently. These include the efficient delayed neutron treatment algorithm [2], the steady-state modelling by the TDMC neutron tracking [3] and the statistical uncertainty estimation by the history-based batch method [4].

The TDMC neutron transport analysis is broadening its applications for nuclear reactors, benefiting from the lack of approximations and leveraging ever-advancing computing power. The transient analysis capabilities using advanced methods have been successfully demonstrated across various MC codes, including McCARD [3–6], TRIPOLI-4 [7], Serpent-2 [8], and RMC [9]. However, the massive computational demand of the TDMC simulation still reveals limitations of its practical usage even on cluster computers with hundreds of central processing unit (CPU) processors.

Currently, computing technology that utilizes graphics processing units (GPUs) is demonstrating significant cost-effective competitiveness compared to traditional multi-core CPU platforms. To fully exploit the single instruction multiple data (SIMD) architecture such as vector computers and GPUs, the MC neutron simulation algorithm must

transition from the conventional history-based approach to an optimized event-based method [10]. Recent developments of GPU-based continuous-energy MC codes [11,12] for the steady-state reactor core analysis have demonstrated speedups of at least a hundredfold on a gaming GPU relative to a single-core performance.

To improve the performance of the TDMC neutron transport analysis of McCARD [13], its new GPU-utilizing version, named McCARD/G, is currently under development. This version features a newly implemented MC neutron tracking module using an event-based algorithm and a generalized lattice geometry treatment module, alongside the adaptation of other major modules to the GPU architecture. Section 2 presents a detailed summary of the newly developed event-based MC algorithm and the optimizations applied for GPU execution. Section 3 presents application results for the C5G7-TD numerical benchmark [14] and the experimental benchmark at the Kyoto University Critical Assembly (KUCA) [15].

2 Algorithms and features

2.1 Event-based TDMC neutron tracking algorithm

The TDMC neutron tracking algorithm in McCARD [3] adopts splitting of the time domain into time bins and the time bin-wise population control [4] by the combing technique [16]. In McCARD/G, the TDMC algorithm has been newly developed by implementing tasks to be executed on GPUs in *kernel* functions of the Compute Unified Device Architecture (CUDA) framework [17]. Figure 1 shows a

* e-mail: shimhj@snu.ac.kr

```

for time bins
  while alive_num > 0
    parallel foreach alive_num neutrons in the current-time-bin source buffer
      for num_repeat
        Calculate distance-to-collision (DTC), distance-to-surface (DTS), and
          distance-to-time-bin-boundary (DTT)
        Select the minimum distance among DTC, DTS, and DTT
        Update the neutron location by the minimum distance
        Update the neutron time by the minimum distance
        if DTT is selected then
          Store neutron for an next time-bin source
          break
        else if DTS is selected then
          Update the current cell as the entering cell
        else if DTC is selected then
          Apply the implicit capture
          Sample a collision type without an absorption reaction
          Simulate the selected reaction with updating the current-time-bin
            neutron buffer and the precursor buffer
        end if
      end for
    end parallel foreach
    Update alive_num
  end while

  parallel foreach number of neutrons in the next-time-bin source buffer
    Comb the stored neutrons for the next-time-bin simulations
  end parallel for
  parallel foreach number of precursors in the precursor buffer
    Comb the stored precursors
    Sample the delayed fission neutrons and append them to the next-time-bin
      source buffer
  end parallel for
  Update the current-time-bin source buffer by the next-time-bin source buffer and
    empty the next-time-bin source buffer
end for

```

Fig. 1. Pseudocode of the event-based TDMC neutron tracking algorithm in McCARD/G.

pseudocode for the McCARD/G's TDMC neutron tracking algorithm. In the figure, each contents inside of the “parallel foreach” loops are implemented in a kernel function which can be called by the CPU running.

As shown in [Figure 1](#), the first kernel function is called while the number of neutrons to be processed in the current time bin, denoted by *alive_num*, is greater than zero. Then, successive sets of the flight and collision events of a neutron as many as *num_repeat* are simulated on GPUs. In this study, the value of *num_repeat* is determined to be

16 from a performance evaluation, written in [Section 3](#). In the neutron's flight simulation, a flight distance is selected as the minimum value among the distance-to-collision (DTC), the distance-to-surface (DTS), and the distant-to-time-bin-boundary (DTT). Then the neutron time can be readily updated by an addition of $L_{\min}/\sqrt{2E_n/m_n}$ where $L_{\min}$, E_n and m_n are the minimum distance, the kinetic energy, and the mass of the neutron, respectively. When the DTT is selected as $L_{\min}$, its simulation is stopped and its data is stored at the next-time-bin source buffer.

When a collision is sampled by the DTC selection, the implicit capture technique is applied prior to the collision simulation, in which the weight bounds is adjusted to be proportional to the average weight of the neutrons in this time bin [4]. After a selection of the collision type, surplus neutrons produced from the fission or multiplicative reaction with their new energy and direction are stored at the current-time-bin source buffer to be processed by the next *kernel* call in the while loop. When a delayed fission neutron is produced exceeding the current time bin during the collision simulation, it should be noted that its simulation for the current time bin is terminated and its precursor data is stored at the precursor buffer.

After finishing the *kernel* calls inside the while loop, the *alive_num* variable is updated to reflect the number of neutrons alive in the current time bin. Upon finishing the while loop, the population controls [4] is applied for the neutrons stored at the next-time-bin buffer and the delayed neutron precursors stored at the precursor buffer. As results of the combing technique, designated numbers of neutrons and delayed neutron precursors at the next-time-bin source buffer and the precursor buffer, respectively, are selected. Especially, after the combing of the delayed neutron precursors, the delayed neutrons are sampled for the next time bin and appended at the next-time-bin neutron source buffer. From Figure 1, one can see that the population controls are implemented by additional kernel functions.

2.2 Generalized lattice geometry treatment

The geometry modeling capability of the original version of McCARD, denoted as McCARD/C hereafter, employs the boundary representation (B-Rep) method to accurately model arbitrarily complex geometries. However, the B-rep approach can seriously deteriorate the geometry handling performance on GPUs because the cell searching for a neutron penetrating a surface requires an iterative checking of neighboring cells sharing the surface whether the neutron is crossing its boundary surface. This iteration may require numerous accesses to the GPU's global memory for complex system geometries, although the CUDA programming guide [17] advises minimizing global memory accesses as much as possible.

In McCARD/G, therefore, special lattice geometry handling capabilities has been newly developed to make the best of the fact that a complex geometry, such as a nuclear reactor core, can be modelled as a hierarchical repetition of simple unit geometries. For instance, the fuel region of a pressurized water reactor core can be modelled using a cuboidal lattice cell that contains a single fuel rod. Then the fuel rod lattice cell can ensure that only one cell sharing a surface exists along the neutron direction, thereby eliminating the need for the iterative cell searching.

In addition to a geometry handling module designated for the repeated structure of cylindrical fuel rods, a generalized lattice treatment module has been developed to facilitate the B-Rep modeling within the unit lattice cell.

This integration merges the flexibility of the arbitrary geometry modeling capability of B-Rep with the efficiency of the lattice structure. The generalized lattice geometry treatment improves performance by enabling efficient cache utilization on GPUs through the reuse of geometric data in the B-Rep modeling and an optimized simulation of neutron movements between lattice cells.

2.3 Other features

The continuous-energy cross section treatment module in McCARD/G has been implemented by porting the cross-section data arrays to GPU, maintaining the same data structure as in McCARD/C. The hashed unionized grid method [12], which retains the benefits of the original unionized grid method [18] while mitigating the memory usage issues associated with the double index, is applied for efficient energy grid searches in McCARD/G. The frequency of the energy grid search becomes reduced by conducting it only after the collision event occurs, rather than with every total cross section calculation as in McCARD/C.

In McCARD/G, the TDMC history-based batch method [4] has been implemented for an accurate estimation of the statistical uncertainty from the TDMC neutron simulations. For this purpose, the batch index is assigned for each neutron while its progeny inherits the index of the parent neutron. Tally means are calculated on a batch-wise basis, and the variance of the final mean value is determined from these batch-wise means. It is important to note that in the TDMC history-based batch method, the population controls are executed batch-wise to ensure independence between batches. The verification results of this method in McCARD/G are detailed in Section 3.

3 Numerical results

3.1 C5G7-TD benchmark

3.1.1 C5G7-TD problem specification

The C5G7-TD benchmark [14] specifies a series of space-time neutron kinetics problems disregarding feedback effects. Figure 2 shows its core configuration, which consists of 16 assemblies, each with 17x17 fuel pins of either UO₂ or MOX, surrounded by a reflector region. The C5G7-TD benchmark provides four exercise sets from 0 to 3 with the two dimensional (2D) configuration, and two exercise sets of 4 and 5 in three dimensional (3D) geometry.

Transient events in the 2D problem sets are modeled through time-dependent adjustments to the seven-group macroscopic cross sections in the mixed region of the control rod and guide tube. For the 3D problems, scenarios involving axial movements of the control rod are provided. In this study, the 2D problems of TD0-1, TD1-1 and TD2-1 are selected with the bank 1, specified in Figure 2, following the scenarios depicted in Figure 3. For the 3D

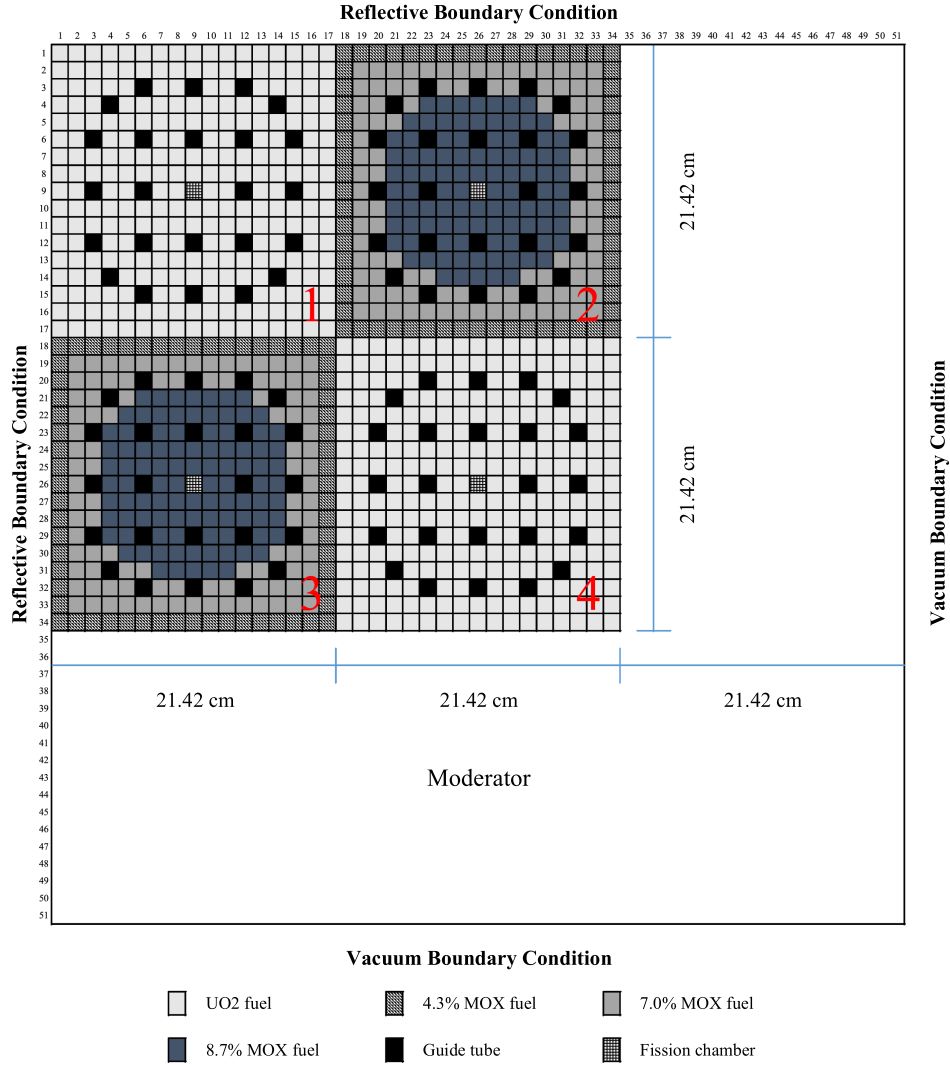


Fig. 2. Core configuration and bank numbers of the C5G7-TD benchmark.

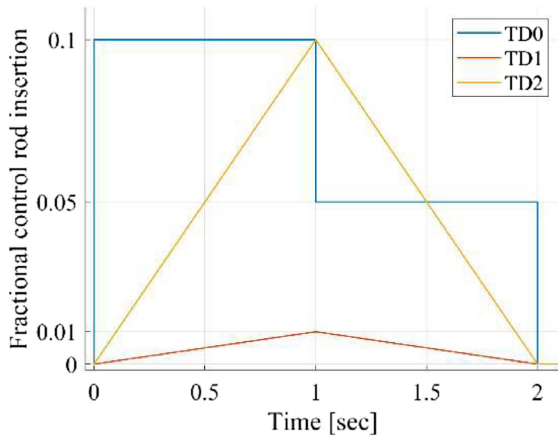


Fig. 3. Transient scenario of the TD0, TD1 and TD2 problems.

problems, TD4-1, TD4-3, and TD4-4 are chosen, with the control rod movement scenarios illustrated in Figure 4.

3.1.2 Determination of TDMC parameters

First, the optimal value of *num_repeat* – the number of successive sets of the flight and collision events – is determined through steady-state TDMC calculations [3] for the C5G7 TD0-1 problem. These calculations are conducted on 500 000 neutron histories per time bin with varying *num_repeat* by a factor of 2 while maintaining a fixed number of neutron histories. Figure 5 displays a trend in the elapsed time per time bin for the McCARD/G calculations. From this data, *num_repeat* of 16, which yields the best performance, is selected for this study.

Next, the accuracy of the TDMC history-based batch method implemented in McCARD/G is verified for the TD0-1 problem. The standard deviation (SD), σ_{HB} , of the core fractional fission rate is compared to the reference SD, σ_{ref} , obtained by the sample SD of mean estimates calculated from independent McCARD/G runs. The TDMC history-based batch method is applied with 50 batches, each containing 10 000 neutrons and 10 000

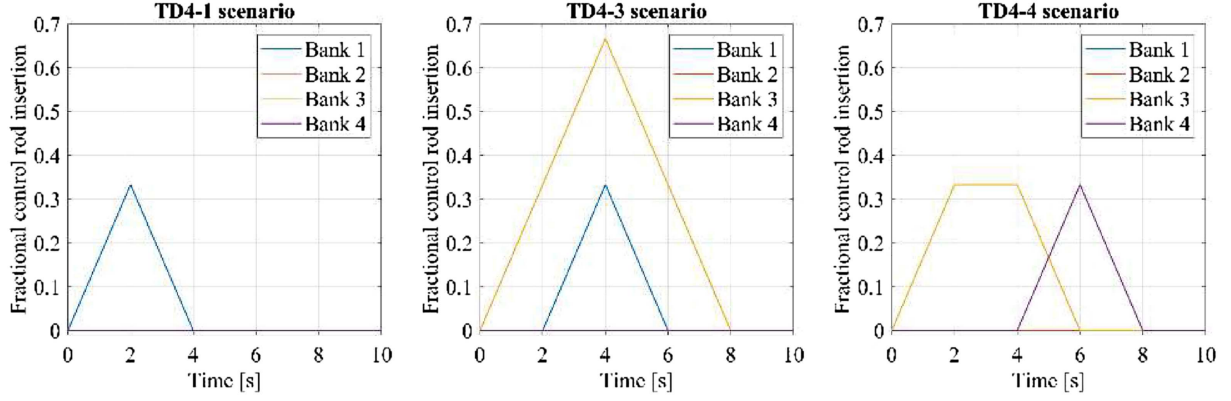


Fig. 4. Transient scenarios of the TD4-1, TD4-3 and TD4-4 problems.

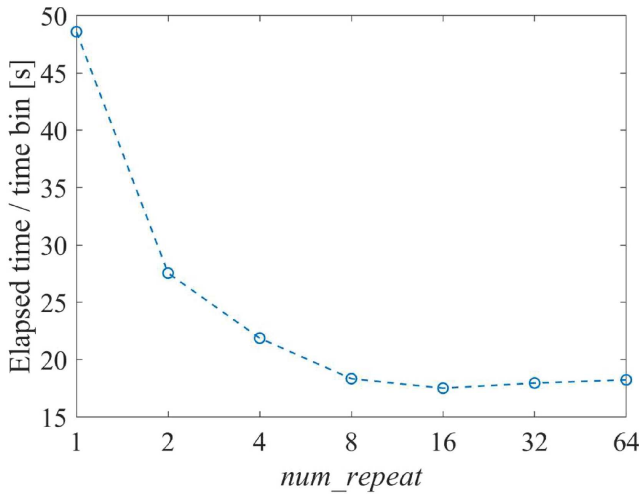


Fig. 5. Elapsed time per time bin according to *num_repeat* for the C5G7 TD0-1 TDMC steady-state calculations.

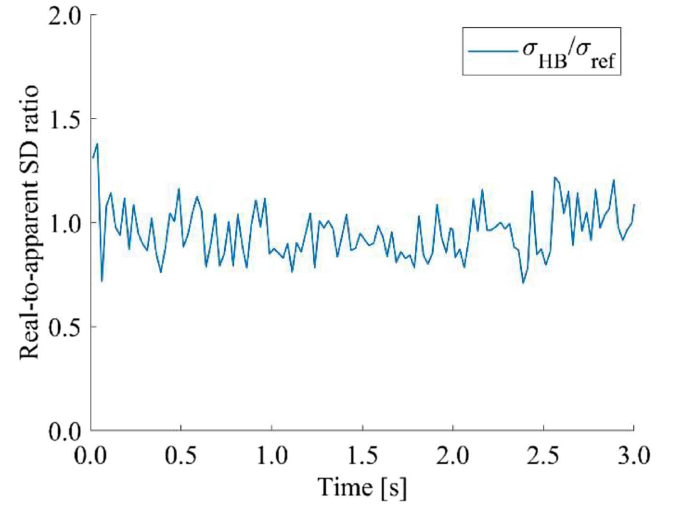


Fig. 6. Ratio of σ_{HB} to σ_{ref} for the core fractional fission rate of the C5G7 TD0-1 problem.

precursors. σ_{ref} is calculated from 50 independent runs of McCARD/G, each with 500 000 neutrons and 500 000 precursors, with the seed of the random number generator varied each time. The time bin size is set to 0.5 ms. Figure 6 shows the ratio of σ_{HB} to σ_{ref} . From the figure, one can see that the ratio oscillates around 1.0, which indicates that σ_{HB} 's agree well with σ_{ref} 's.

3.1.3 Numerical results

The McCARD/G TDMC calculations for the C5G7-TD problems are conducted using 50 000 neutrons and 50 000 precursors per time bin, set to 0.5 ms, with the option of 50 batches. For the TDMC steady-state calculations, 100 convergence steps and 500 precursor generation steps [4] are applied. The McCARD/G calculations are conducted on five NVIDIA GeForce RTX 4090 GPUs. McCARD/G TDMC results are verified by comparing with those calculated by the deterministic transport analysis code, nTRACER [19].

Figure 7 presents comparisons of the core dynamic reactivity and the fractional fission rates for the C5G7 TD 2D problems of TD0-1, TD1-1, and TD2-1. In the

McCARD/G results, the error bars represent the 95% confidence interval. On the left side figure, the dynamic reactivity of TD0-1 exhibits a sharp decline and rise due to the stepwise insertion and withdrawal of control rods, whereas TD1-1 and TD2-1 display linear changes in reactivity corresponding to the control rod movements. From the figures, it is shown that the McCARD/G results agree well with nTRACER's within their error ranges.

Figure 8 shows comparisons of the dynamic reactivity and the fractional fission rate of the TD4-1, TD4-3, and TD4-4 problems calculated by McCARD/G and nTRACER. The figures demonstrate that the results from McCARD/G agree well with those from nTRACER within the 95% confidence intervals, as the accuracy of the CPU version of McCARD [4].

In order to compare the performance of McCARD/G with the original CPU version of McCARD, the McCARD TDMC calculations are conducted using 96 Intel Xeon X5650 CPU processors for the TD0-1 and TD4-1 problems. The elapsed time comparisons reveal that the speedups achieved by a single GPU card are 238 times for TD0-1 and 261 times for TD4-1, respectively.

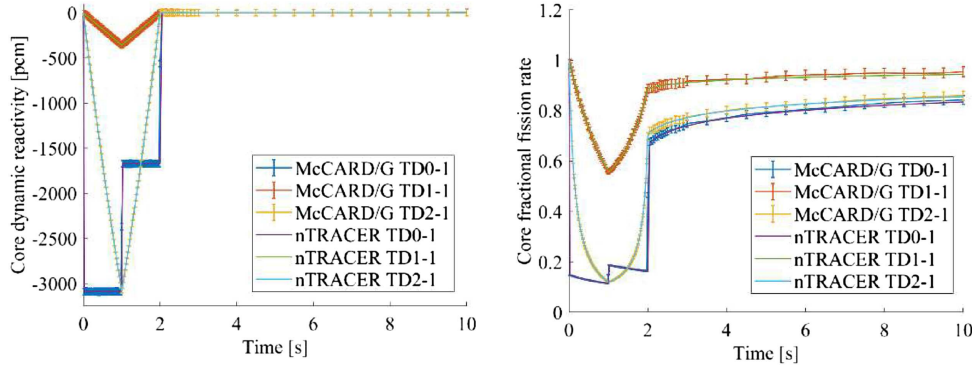


Fig. 7. Core dynamic reactivity and core fractional fission rate for TD0-1, TD1-1 and TD2-1.

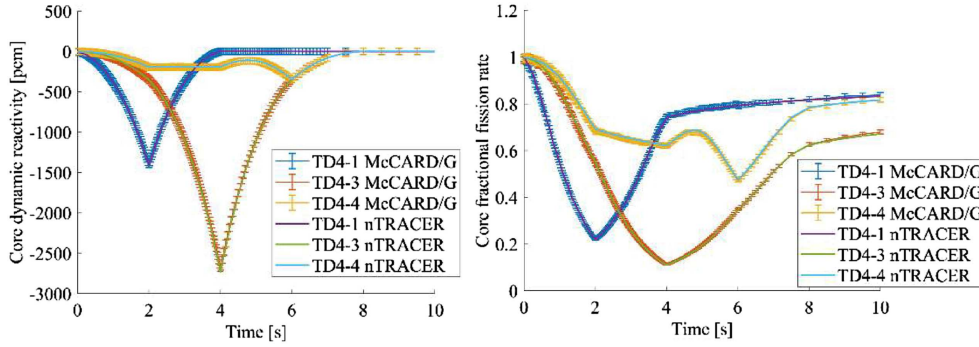


Fig. 8. Core dynamic reactivity and core fraction fission rate for the TD4 problems.

3.2 KUCA experimental benchmark

The McCARD/G TDMC calculations using the continuous-energy cross section libraries are applied for a numerical simulation of the prompt neutron decay constant, α , measurement by the pulsed neutron source (PNS) experiment in the accelerator driven system (ADS) benchmark [15]. The benchmark provides six different subcritical cores comprised of Pb–Bi loaded enriched uranium fuel and polyethylene moderator and reflector. Among the six core configurations of subcritical states, case 6 is solved in this analysis. Figure 9 shows the core configuration of case 6. The core consists of plate-type, 93% enriched uranium fuel assemblies with partially inserted Pb–Bi plates, along with a polyethylene moderator and reflector, as shown in Figure 10. Spallation neutrons are generated by a 100 MeV proton beam injected into the Pb–Bi target at the center of the core.

The McCARD/G PNS simulation is conducted in a time bin size of 0.1 ms on 10 000 000 neutrons in total per time bin with 1000 batches having the batch size of 10 000. Note that the TDMC simulations are conducted only considering the prompt neutrons. The angular distribution and spectra of spallation neutrons are obtained from the previous research [5].

Figure 11 shows the time-dependent neutron density values, with their 95% error bars, tallied at three detectors, presented in Figure 9. From the figure, one can see that the detector signals show different profiles depending on their positions. From these time-dependent tally

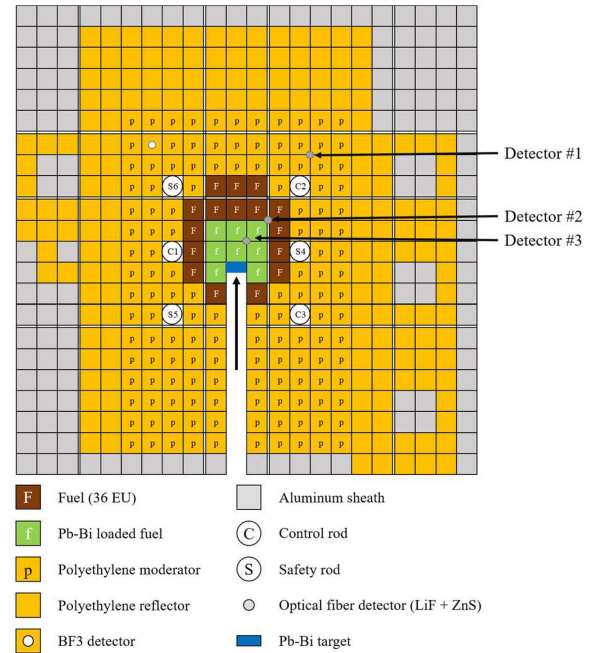


Fig. 9. Core configuration of the KUCA ADS benchmark case 6 problem.

results, α can be estimated by the exponential fitting with changing the start time of the fitting interval. Figure 12 shows comparisons of the α value estimated by the expo-

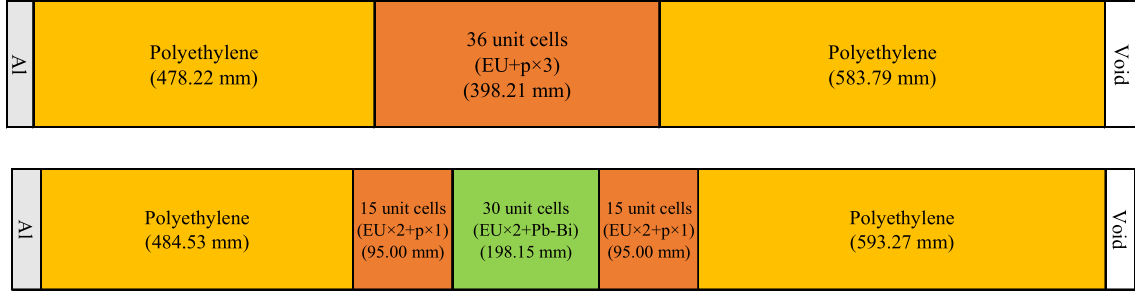


Fig. 10. Side view of enriched uranium fuel assembly (upper, “F” in Fig. 9) and Pb-Bi loaded fuel assembly (lower, “f” in Fig. 9).

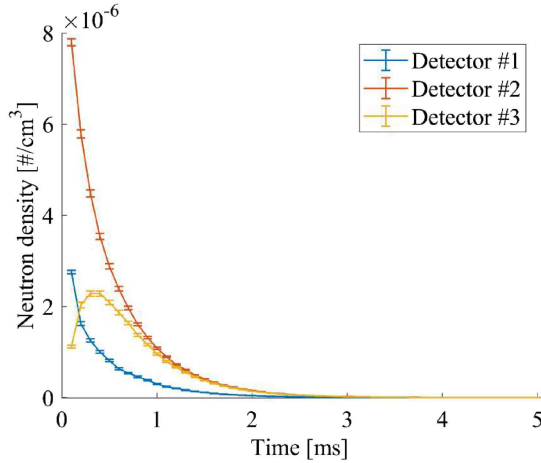


Fig. 11. Time-dependent neutron densities over time for the KUCA ADS benchmark case 6 problem.

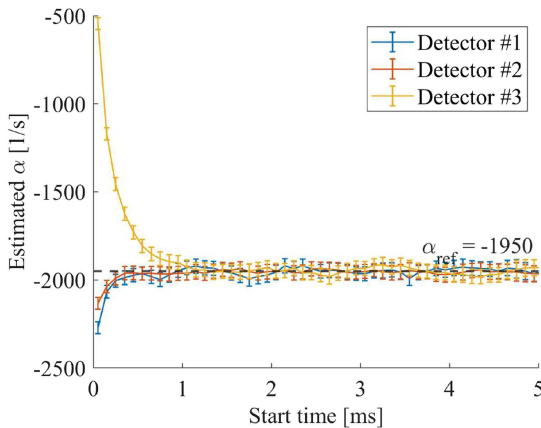


Fig. 12. Convergence trend of estimated α value at different detector positions for the KUCA ADS benchmark case 6 problem.

nential fitting with a reference value calculated by the α iteration method [20] using McCARD. From the figure, one can see that the estimated α in the three detector positions show different convergence trends according to the fitting start time, but finally converge to the reference value of 1950 s^{-1} .

4 Conclusion

The GPU-enhanced version of McCARD, named McCARD/G, is currently under development with a focus on accelerating the TDMC neutron transport analysis. In McCARD/G, an event based neutron tracking algorithm has been newly developed for the TDMC simulation. Additionally, a generalized lattice geometry treatment capability has been implemented, which enhances rapid cell search and precise geometry modeling by integrating the B-Rep modeling capability with the lattice structure. The TDMC history-based batch method, essential for accurately quantifying the real variance from the TDMC calculations, has been successfully implemented and verified. The TDMC capabilities of McCARD/G have been verified through the C5G7 TD benchmark problems, with performance enhancements measuring over 200 times per GPU card compared to the original CPU version. Moreover, the integrity of the continuous energy cross section treatment and tally capabilities has been confirmed through the KUCA ADS benchmark.

Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2022M2D2A1A02063865).

Conflicts of interest

The authors declare that they have no competing interests to report.

Data availability statement

No data are associated with this article.

Author contribution statement

McCARD/G has been developed by Seong Jeong Jeong, Woo Kyoung Ko, and Young In Kim under the supervision of Hyung Jin Shim. The numerical analyses have been conducted by Woo Kyoung Ko and Young In Kim. The manuscript has been written by Woo Kyoung Ko, Young In Kim, and Hyung Jin Shim.

References

1. E.L. Kaplan, *Monte Carlo Methods for Equilibrium Solutions in Neutron Multiplication*, UCRL-5275-T (Lawrence Livermore National Laboratory, Washington, DC, 1958)

2. B.L. Sjenitzer, J.E. Hoogenboom, Dynamic Monte Carlo method for nuclear reactor kinetics calculations, *Nucl. Sci. Eng.* **175**, 94 (2013)
3. N. Shaukat, M. Ryu, H.J. Shim, Dynamic Monte Carlo transient analysis for the Organization for Economic Co-operation and Development Nuclear Energy Agency (OECD/NEA) C5G7-TD benchmark, *Nucl. Eng. Technol.* **49**, 920 (2017)
4. S.H. Jang, H.J. Shim, Advances for the time-dependent Monte Carlo neutron transport analysis in McCARD, *Nucl. Eng. Technol.* **55**, 2712 (2023)
5. S.H. Jang, H.J. Shim, Determination of an effective detector position for pulsed-neutron-source alpha measurement by time-dependent Monte Carlo neutron transport simulations, *Sci. Technol. Nucl. Ins.* **2018**, 2350458 (2018)
6. J.M. Park, C.H. Pyeon, H.J. Shim, Sensitivity of a control rod worth estimate to neutron detector position by time-dependent Monte Carlo simulations of the rod drop experiment, *Nucl. Eng. Technol.* **56**, 916 (2024)
7. M. Faucher, D. Mancusi, A. Zoia, New kinetic simulation capabilities for Tripoli-4®: Methods and applications, *Ann. Nucl. Energy* **120**, 74 (2018)
8. V. Valtavirta, M. Hessian, J. Leppänen, Delayed neutron emission model for time dependent simulations with the serpent 2 Monte Carlo Code – first results, in *Proceedings of PHYSOR 2016* (2016)
9. C. Jia, L. Jian, X. Guo, K. Wang, S. Huang, J. Liang, Development of an improved direct kinetic simulation capability in RMC code, *Ann. Nucl. Energy* **173**, 109110 (2022)
10. F.B. Brown, W.R. Martin, Monte Carlo methods for radiation transport analysis on vector computers, *Prog. Nucl. Energy* **14**, 269 (1984)
11. S.P. Hamilton, T.M. Evans, Continuous-energy Monte Carlo neutron transport on GPUs in the Shift code, *Ann. Nucl. Energy* **128**, 236 (2019)
12. N. Choi, K.M. Kim, H.G. Joo, Optimization of neutron tracking algorithms for GPU-based continuous energy Monte Carlo calculation, *Ann. Nucl. Energy* **162**, 108508 (2021)
13. H.J. Shim, B.S. Han, J.S. Jung, H.J. Park, C.H. Kim, McCARD: Monte Carlo code for advanced reactor design and analysis, *Nucl. Eng. Technol.* **44**, 161 (2012)
14. V.F. Boyarinov, et al., Deterministic time-dependent neutron transport benchmark without spatial homogenization (C5G7-TD), NEA/NSC/DOC(2016), OECD Nuclear Energy Agency, Version 1.9, 2018
15. C.H. Pyeon, Experimental benchmarks of neutronics on solid Pb–Bi in accelerator-driven system with 100 MeV protons at Kyoto University Critical Assembly, KURRI-TR-447, Research Reactor Institute, Kyoto University, 2017
16. T.E. Booth, A weight (charge) conserving importance-weighted comb for Monte Carlo, LA-UR96-0051, Los Alamos National Laboratory, NM (United States), 1996
17. NVIVIA CUDA Programming Guide 2.2, 2009
18. J. Leppänen, Two practical methods for unionized energy grid construction in continuous-energy Monte Carlo neutron transport calculation, *Ann. Nucl. Energy* **36**, 878 (2009)
19. Y.S. Jung, C.B. Shim, C.H. Lim, H.G. Joo, Practical numerical reactor employing direct whole core neutron transport and subchannel thermal/hydraulic solvers, *Ann. Nucl. Energy* **62**, 357 (2013)
20. H.J. Shim, S.H. Jang, S.M. Kang, Monte Carlo alpha iteration algorithm for a subcritical system analysis, *Sci. Technol. Nucl. Ins.* **2015**, 859242 (2015)

Cite this article as: Woo Kyoung Ko, Seong Jeong Jeong, Young In Kim, and Hyung Jin Shim. Development of a GPU-enhanced time-dependent Monte Carlo neutron transport version of McCARD, *EPJ Nuclear Sci. Technol.* **10**, 28 (2024)