Development Status of a GPU-based Monte Carlo Code GRAMS for Nuclear Reactor Analysis

Ao Zhang[a,b], Ser Gi Hong[a*], Xiangzhou Cai[b]
[a]*Department of Nuclear Engineering, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul, Korea*
[b]*State Key Laboratory of Thorium Energy, Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai, China*
[*]*Corresponding author: hongsergi@hanyang.ac.kr*

Keywords : GPU Monte Carlo, PLCG, hybrid tracking, persistent thread, thermal and fast reactors

1. Introduction

Monte Carlo particle transport method owns high fidelity in nuclear reactor physics analysis by adopting continuous-energy nuclear data and explicitly modeling complex geometry. With the rapid evolution of GPU, accelerating Monte Carlo particle transport simulations on modern GPU architectures becomes promising. GRAMS (**G**PU-based **R**eactor Physics **A**nalyzer with **M**onte Carlo **S**imulation) is under development at Hanyang University for efficient high-fidelity thermal and fast reactor analysis on Nvidia GPU platforms. This paper introduces the development status of GRAMS, including neutron physics, geometry, hybrid surface and delta tracking, history-based parallel transport algorithm. Additionally, the code-to-code verification results of GRAMS against OpenMC for fresh-fuel criticality are given for a pressurized water reactor, a sodium-cooled fast reactor, and a molten salt reactor, showing good accuracy and efficiency of GRAMS in thermal and fast reactor analysis.

2. Overview of GRAMS

2.1 Neutron physics

The ACE format continuous-energy nuclear data are utilized in GRAMS for neutron transport simulations, including the data from both free gas model and thermal scattering law (TSL). GRAMS organizes ACE data into six distinct structures, including temperature-dependent cross sections, zero-Kelvin elastic scattering cross sections for Doppler broadening rejection correction (DBRC), fission yields, angle and energy distributions, unresolved resonance probability tables, and TSL data. All of them adopt a structure of array layout for improving GPU throughput by increasing coalesced memory accesses. The hash-based equal-lethargy energy grid method [1] is used with 2^{14} energy bins to greatly reduce energy grid search range by O(1) search for subsequent binary search, thereby accelerating cross section evaluation.

Target thermal motions are processed in GRAMS using three sampling methods, i.e., the constant cross section approximation (CXS), the DBRC, and the Relative Speed Tabulation (RST) [2,3]. While the DBRC applies a rejection sampling to the CXS sampled target velocity, the RST samples the relative velocity between a neutron and the target using precomputed cumulative distribution tables, thereby featuring better efficiency than the DBRC. In GRAMS, the DBRC and RST are applied by default within the energy range of 0.1-210 eV for selected heavy nuclides. Meanwhile, the CXS is utilized for all nuclei when the neutron energy falls below 400 times the kinetic energy of target.

Fig. 1 presents the sampled outgoing neutron energy distribution for an incident neutron with energy of 20.25 eV and the target nucleus of U-238 at 600 K. The DBRC and RST methods achieve excellent agreements while the CXS method underestimates up-scatterings resulted from U-238 thermal motions. The elastic scattering cross sections near resonance peaks vary sharply, making the CXS assumption invalid and requiring the use of DBRC or RST method.

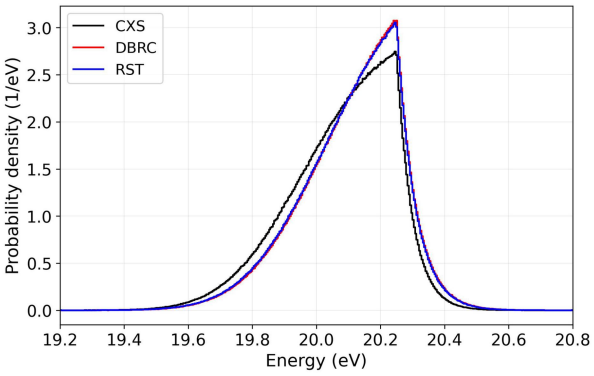


Fig. 1. Probability densities of outgoing neutron energy sampled using the CXS, the DBRC, and the RST methods. The incident neutron energy is 20.25 eV and the target nucleus is U-238 at 600 K.

2.2 Geometry and visualization

Considering that most commercial reactors are designed with three levels, i.e., pin cell, assembly, and core, a *Primitive-Lattice-Core* Geometry (PLCG) [4] method was proposed and implemented in GRAMS for modeling rectangular and hexagonal reactor geometries while maintaining computational efficiency on the GPU. The PLCG uses the predefined constructive solid geometry (CSG) based rectangular or hexagonal *primitives* to accurately represent axially infinite 2D radial pin cells, leveraging their inherent concentric symmetry. The 2D *primitives* are then organized into *lattice*, which is a 3D rectangular or hexagonal array.

Every *lattice* has its own axial grid, which is particularly well-suited for modeling fuel assemblies with different control rod insertions. Different from *lattice*, *core* is a 2D array of *lattice* without axial structures. Additionally, *lattice bounds* and *core bounds* are designed to represent assembly gaps or ducts as well as core baffles or barrels.

GRAMS leverages the C++ PLplot library to render cross-sectional slices for geometry diagnostics and the VTK XML Rectilinear Grid (VTR) format to visualize spatial tally distributions. The whole geometry is divided into pixels for the former and voxels for the latter, the GPU-based PLCG tracker is then called to retrieve corresponding material or tally data for visualizations. Fig. 2 presents geometry visualizations for the VERA fuel assembly, the MSRE (Molten Salt Reactor Experiment) fuel assembly, and the CEFR (China Experimental Fast Reactor) fuel and safety sub-assemblies (SAs), showing geometry capabilities of GRAMS for modeling thermal and fast reactors.

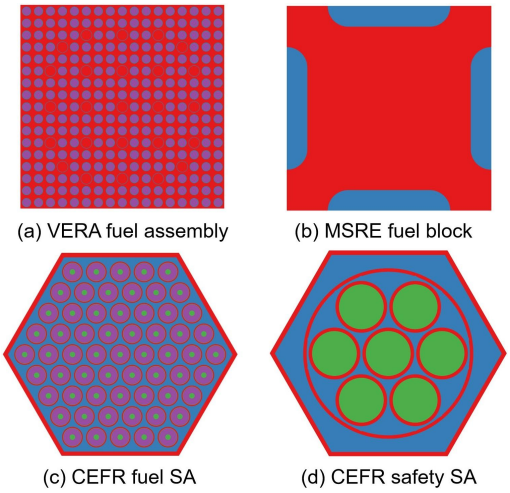


Fig. 2. GRAMS visualizations for the VERA and the MSRE fuel assemblies, and the CEFR fuel and safety SAs.

2.3 Hybrid surface and delta tracking

GRAMS supports three tracking routines which can be specified in the input file. We implemented a hybrid surface and delta tracking scheme in GRAMS since the conventional surface-tracking method encounters significant performance loss when the neutron mean free path is long compared to geometric dimensions. The hybrid tracking scheme partitions the geometry into a global delta-tracking domain and the left surface-tracking regions, as shown in Fig. 3. The surface-tracking regions are PLCG components containing highly absorbing materials that would otherwise dominate the global majorant cross sections [5], thereby making the global majorant cross section in delta-tracking domain relatively tight. In GRAMS, the majorant cross section is constructed on the equal-lethargy energy grids, which are adopted for accelerating energy grid lookup.

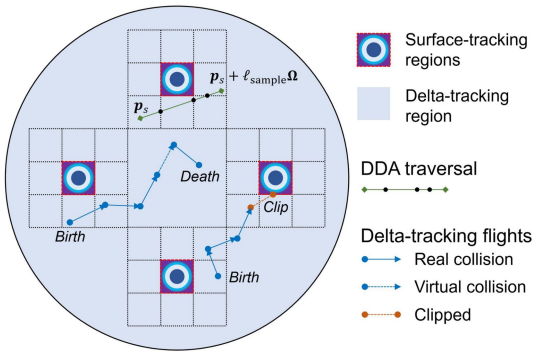


Fig. 3. Illustration of the hybrid surface and delta tracking scheme with highly absorbing materials.

The hybrid tracking scheme uses the PLCG component flags for dispatching surface-tracking or delta-tracking routine on the GPU, which requires no additional cross section evaluation compared to the switching criterion adopted in the CPU-based Monte Carlo code Serpent [5]. In commercial reactors, most geometry components are processed by the delta tracking in the hybrid scheme, eliminating a large fraction of boundary-distance calculation costs compared to surface tracking.

As seen in Fig. 3, a particle is transported directly to the candidate collision position if the track segment from $\boldsymbol{p}_s$ to $\boldsymbol{p}_s+l_{sample}\Omega$ does not intersect a surface-tracking region. Otherwise, the flight is clipped at the first entry boundary of the surface-tracking region, and subsequent transport is performed using surface tracking. A complete particle history may therefore consist of a sequence of real and virtual candidate collisions, interspersed with transitions between the two tracking modes, until the particle history is terminated. To efficiently detect delta-tracking flight intersections with surface-tracking regions, a linear digital differential analyzer (DDA) algorithm is adopted based on the PLCG core or lattice arrays.

2.4 Persistent-thread history-based algorithm

The hybrid surface and delta tracking scheme was implemented into the persistent-thread history-based tracking algorithm, as described in Algorithm 1. The persistent-thread algorithm maintains a fixed number of resident CUDA threads, each of which immediately fetches an unprocessed particle from the source bank once the processing particle on the thread is terminated. Consequently, this algorithm minimizes idle threads without compacting active and dead particles during transport, which is the core idea of event-based parallel transport algorithms.

A complete particle history in the hybrid tracking scheme is outlined in Algorithm 1, where surface tracking and delta tracking share the same collision physics sampling routines. The tracking routine to be adopted can be determined by searching the material index and the hierarchical PLCG indices at which a

particle resides. For the delta-tracking routine, the free-path length l_{sample} is sampled using the majorant cross section and the distance to the nearest limiting boundary d_{lim} is evaluated by the DDA traversal. The particle is advanced by min (l_{sample}, d_{lim}), and boundary crossing or candidate collision are then processed. Both real and virtual collisions are scored using the candidate collision estimator, and the execution is directly back to the line 4 in Algorithm 1 for virtual collisions. Otherwise, the collision physics is sampled for real collisions. For simplicity, explanation of the conventional surface-tracking routine is omitted.

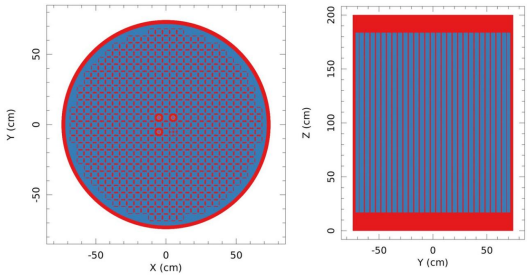
Algorithm 1 History-based hybrid surface and delta tracking algorithm

```
1:  for all threads in parallel do
2:      while source bank 𝒮 is not exhausted do
3:          Fetch one particle p from 𝒮
4:          while p is alive do
5:              Determine material and hierarchical geometry indices
6:              if p ∈ delta-tracking region then
7:                  Calculate ℓ_sample using majorant XS and d_lim
8:                  Move p by min(ℓ_sample, d_lim)
9:                  if ℓ_sample ≥ d_lim then
10:                     Processing tracking or geometry boundary; continue
11:                 else
12:                     Locate the material at the candidate collision site
13:                     Evaluate macroscopic cross sections
14:                     Score tallies using the candidate collision estimator
15:                     if ξ′ ≥ Σ_t/Σ_maj then continue
16:             else
17:                 Evaluate macroscopic cross sections
18:                 Calculate ℓ_sample and d_surf
19:                 Move p by min(ℓ_sample, d_surf)
20:                 Score track-length tallies over the traveled segment
21:                 if ℓ_sample ≥ d_surf then
22:                     Process geometry surface or boundary; continue
23:             Sample the real collision and update the particle state
24:             Sample fission sites into fission bank when applicable
```

3. Results and discussion

To demonstrate the thermal and fast reactor analysis capabilities of GRAMS, three mockup core models based on the VERA benchmark, the CEFR benchmark, and the MSRE are adopted for verification by comparing against OpenMC. The three mockup models retain most of the radial heterogeneity while simplify the axial structures, reducing the geometry modeling complexity for code-to-code comparisons. All of the three models contain highly absorbing materials for testing the hybrid tracking scheme, such as poison rods, safety rods, or control rods. The mockup core models for the VERA and CEFR benchmarks are given in the reference [4], and the MSRE mockup core is presented in Fig. 4, consisting of active core, radial down-comer, and axial lower and upper plenums. The three models have control rods fully inserted.

We simulated the three models using OpenMC on a single Intel Core Ultra 7 270K Plus CPU and GRAMS on a single Nvidia RTX 4090 GPU with same particle histories. For the VERA and MSRE models, resonance scattering treatments and TSL data were adopted. The VERA model was simulated using 500 inactive and active cycles with five million particles per cycle. For the CEFR and MSRE models, we adopted 150 inactive cycles and 350 active cycles with one million particles per cycle. Radial and axial power distributions were tallied for comparisons.



(a) Horizontal slice (b) Vertical slice

Fig. 4. Horizontal and vertical layout of the MSRE.

The simulation results are summarized in Tables I and II. The radial and axial power differences are visualized in Fig. 5. The k_{eff} differences for the three problems are smaller than 20 pcm, and the root mean square errors in the pin or block power are within 0.3%. These results show excellent agreements between two codes considering the existence of tally uncertainties. Besides, GRAMS is 16-20 times faster than OpenMC on their corresponding consumer-grade GPU and CPU.

Table I. Comparison for criticality eigenvalue and time usage of OpenMC and GRAMS.

Problem	k_{eff}		Time (s)	
	OpenMC	GRAMS	OpenMC	GRAMS
VERA	0.98505 (1)	0.98523 (1)	66166	3236
CEFR	0.94125 (4)	0.94120 (3)	17423	973
MSRE	0.92929 (6)	0.92943 (5)	4845	301

Table II. Statistics of VERA and CEFR pin power differences and MSRE block power difference.

Problem	MIN	MAX	RMS
VERA	-1.203%	1.189%	0.279%
CEFR	-0.673%	0.693%	0.161%
MSRE	-0.334%	0.399%	0.114%

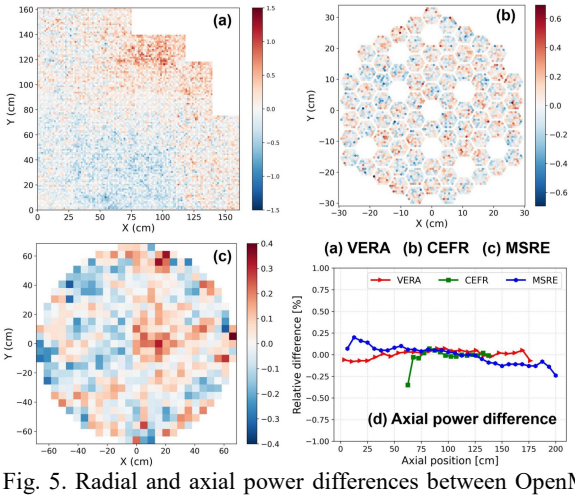


Fig. 5. Radial and axial power differences between OpenMC and GRAMS for the VERA, CEFR, MSRE problems.

The radial and axial normalized volume-integrated power distributions for the three models are presented in Fig. 6. The radial distributions are output from GRAMS in VTR format and then visualized by ParaView. The VERA model runs with reflective boundaries by utilizing its one-eighth symmetry. The center hexagonal pin cell in the CEFR model is plotted since it contains Cf-252 as a neutron source. For the MSRE model, the square blocks of control rods and sample basket have much larger fuel salt volumes than fuel channels, thereby showing very large normalized powers. These results demonstrate that GRAMS has good tally and visualization capabilities based on the PLCG module.

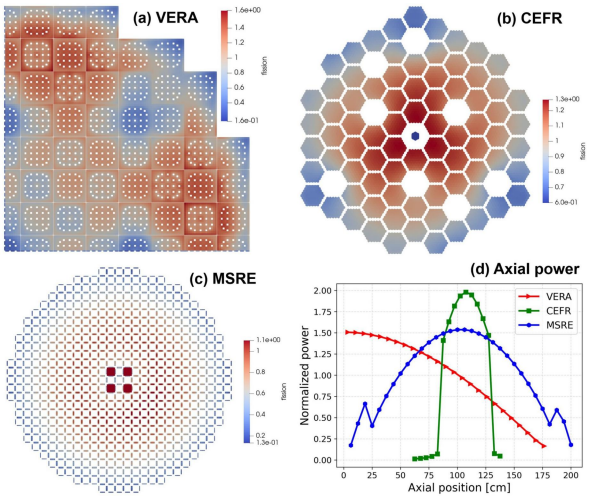


Fig. 6. Radial and axial power distributions generated from GRAMS for the VREA, CEFR, and MSRE models.

The history-based parallel algorithm features high register pressure on the GPU, while the hybrid tracking scheme requires more registers than pure surface or delta tracking due to the use of two free-path flight tracking routines. This results in more local variables spilling to global memory, thereby degrading transport kernel performance. The history-based hybrid tracking algorithm performs more efficient than surface tracking when boundary-distance computational costs dominate the whole simulation time. The CEFR features the long neutron mean free path and compact core layout, and the MSRE fuel block is divided into 10x10 primitives with a side length of 0.508 cm for each primitive. Consequently, both the CEFR and the MSRE run faster using hybrid tracking than surface tracking. However, the surface tracking is slightly faster than the hybrid tracking for the VERA problem.

In the hybrid tracking scheme, the acceptance efficiency of delta tracking is fundamentally bounded by the cross section heterogeneity within a pin cell. Fig. 6 compares the majorant cross sections for the three models, showing large resonance cross sections of structural materials in the CEFR. The MSRE fuel block contains only fuel salt and graphite with relative small cross section differences. As a result, the MSRE has an average candidate collision acceptance probability of 62.1%, which is 29.6% for the CEFR.

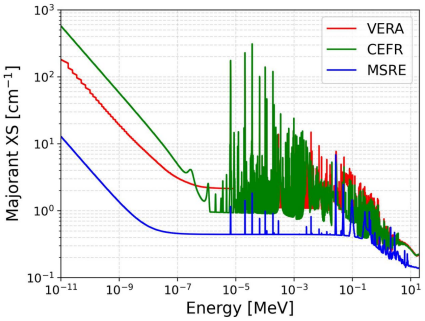


Fig. 7. Majorant cross sections for the three models.

4. Conclusions

This paper presents the development status of GPU Monte Carlo code GRAMS for thermal and fast reactor analysis. GRAMS adopts the continuous-energy nuclear data for neutron collision physics sampling, the PLCG for rectangular and hexagonal reactor core modeling, the hybrid tracking scheme for simulating particle random work, and the persistent-thread history-based algorithm for parallel particle transportation. GRAMS has been verified against OpenMC using the 3D VERA, CEFR, and MSRE core models, showing excellent agreements in k_{eff}, axial and radial high-resolution power distributions. Generally, GRAMS is 16-20 times faster than OpenMC on their corresponding consumer-grade GPU and CPU.

ACKNOWLEDGMENTS

This work is financially supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-02310831) and the National Key Research and Development Program of China (No. 2023YFA1606604).

REFERENCES

[1] Brown, F. B., 2014. New hash-based energy lookup algorithm for Monte Carlo codes (No. LA-UR-14-27037). Los Alamos National Laboratory (LANL).
[2] Zhang, A., Choi, G. Y., Hong, S. G., et al., 2026. An energy and angle sampling program of secondary neutrons based on continuous-energy nuclear data. Transaction of the Korean Nuclear Society Spring Meeting. Jeju, Korea.
[3] Choi, N., Joo, H. G., 2021. Relative speed tabulation method for efficient treatment of resonance scattering in GPU-based Monte Carlo neutron transport calculation. Nuclear Science and Engineering, 195(9), 954-964.
[4] Zhang, A., Hong, S. G., Cai, X. Z. GRAMS: A GPU accelerated continuous-energy Monte Carlo neutron transport code for thermal and fast reactor analysis. Available at SSRN: http://dx.doi.org/10.2139/ssrn.7094480
[5] Leppänen, J., 2010. Performance of Woodcock delta-tracking in lattice physics applications using the Serpent Monte Carlo reactor physics burnup calculation code. Annals of Nuclear Energy, 37(5), 715-722.