Reduced Precision Ray Tracing For Monte Carlo Simulations

Chris Marshall

chrismarshall@uchicago.edu

Introduction

Why do we care about reduced precision ray tracing?

The Precision Paradigm

- Double precision has been the standard for most scientific codes like OpenMC
  - This choice is cautious
    - Errors that can arise from lower precision calculations are hard to detect and mitigate
    - Double precision seems to provide suitable results for almost if not all applications
- However, shifts in computing architectures put double precision in danger
  - GPUs favor lower precision arithmetic that better suits AI workloads
  - Scientific computing relies on GPUs and must adapt to these shifts
- Little is known about the impacts of lower precision arithmetic on Monte Carlo Transport Codes

Related Work

What reduced precision ray tracing work has been done?

Precision in Monte Carlo Transport Codes

- Precision is critical for results, but has rarely been studied
- Most Monte Carlo codes default to using double precision
  - The general community assumption is that this is good enough for our results
- There are a handful of studies investigating the use of single precision in Monte Carlo Transport, but these all have some limitations

Previous Results

- Nelson (2009) developed a basic Monte Carlo code to test the feasibility of utilizing GPUs
  - Nelson noted performance benefits from the use of single precision on simple models, but did little verification of the results
- Lippuner & Elbakri (2011) focused on the performance
  - Lippuner & Elbakri focused on the performance benefits of single precision and provided no baseline comparison to double precision
- The PRAGMA code is a mixed precision Monte Carlo code
  - The novel contributions of the PRAGMA code are based on the specialized geometry system that PRAGMA uses
  - Such contributions cannot be easily generalized to OpenMC
- Liu (2022) investigated single precision in the context of low power devices
  - Like earlier works, little was done to verify single precision results

Open Questions and Research Gaps

- Existing work had done little to address the fundamental question:
  - Can single-precision be reliably used for ray tracing in full-featured Monte Carlo codes?
- Existing studies
  - Use simplified codes and models
  - Lack tests on complex reactor models
  - Lack sweeping analysis across several model types
  - Provide little insight into catastrophic failures that might occur

Finite Precision Ray-Tracing

What can go wrong?

Stuck Particles

- Consider the following scenario:
  - A surface: $x - 1 = 0$
  - A starting position: $x_0 = 1 - \varepsilon$
  - A direction: $(u_x, u_y, u_z) = (1,0,0)$
- The distance to the surface is given by: $d = \frac{1-x_0}{u_x} = \varepsilon$
- If epsilon is too small to represent, the particle does not move
- Since this happens repeatedly, the particle becomes trapped at the same location

Differences between Single and Double Precision

- The events that happen to a particle are important, and differing levels of precision can trigger different events in the same scenario
- To illustrate this, consider the following:
  - A ray: $r(d) = \vec{r_0} + d\vec{u} = \left(-1, r_{0,y}\right) + d(1, 0)$
  - The unit circle: $x^2 + y^2 = 1$
- Solving for the distance from our initial position to the unit circle gives us a quadratic in d
- Changes in the discriminant changes the number of solutions which then changes the number of collisions detected

Differences between Single and Double Precision

Precision	$r_{0,y}$	Discriminant
single (float32)	1.0	0.000000e+00
single (float32)	1.0	0.000000e+00
double (float64)	$1.0 + 10^{-8}$	8.000000e-08
double (float64)	$1.0 - 10^{-8}$	$-8.000000e - 08$

Methodology

How did we investigate reduced precision ray tracing in OpenMC?

Why use OpenMC?

OpenMC has…

- A rich set of problems
  - The set of problems includes validated benchmarks and reactor models
- Geometric complexity
  - OpenMC supports hierarchical universes, lattices, and a wide variety of surface types
- Benchmarking tools
  - OpenMC provides built-in infrastructure to measure performance and results
- Physical tally support
  - OpenMC provides an extensible mechanism for user-defined tallies, allowing for investigation into many physical results

Implementing Variable Precision

- OpenMC was designed and implemented in double precision
  - This means that any support for single-precision would have to modify major parts of the existing codebase
- For ease of testing, we introduced a compile-time constant to toggle precision
- We refactored geometry-related code to use this compile-time constant
  - Decoupling geometric primitives from other primitives is not always clear and can have unintended side-effects
- Ultimately, we used all of this to enable the ability to build OpenMC in either single precision or double precision

Mixed Precision Approaches

- A mixed precision approach to ray-tracing would allow us to compute precision-sensitive pieces of code in double precision and the rest in single precision
  - In many scientific computing disciplines, mixed precision has been an effective means of using single precision in most areas of a code base
  - Determining what pieces of code are precision-sensitive is not trivial
- Design conflicts arise when decoupling pieces of OpenMC from one another
  - This leads to subtle type mismatches that are hard to identify

Test Suite Design

- The core of our test suite was based on the ICSBEP benchmark with the addition of several common reactor problems
  - This totaled 359 problems
  - Includes both simple and complex geometries for a wide range of test coverage
- In designing this test suite, we focused on realistic usage, not problems designed to be hard

While not our focus, a test suite that is designed to be hard is an equally important contribution to the understanding of precision in OpenMC.

Types of Simulation Runs

- Small Test Suite
  - Designed to be a quick verification of the correctness of our implementation
- Varied Convergence
  - Simulated problems at 8 levels of convergence
  - Found that when problems disagree, they do so early on
- Low Uncertainty Suite
  - Simulated problems to high levels of convergence to reveal subtle differences between precision levels
- Reactor-Only Suite
  - Simulated complex problems separately for debugging and analysis purposes
- Complete Suite
  - Simulated all 359 problems in both precision levels

Timing and Performance Tests

- Measured the runtimes of the small modular reactor problem across different CPU platforms to quantify real-world speedups

Platform Name	Number of Cores	CPU Architecture	CPU Platform
MacBook Air	8	ARMv8.4-A	Apple M1
GCP E2	24	x86_64	AMD Rome
GCP C4	8	x86_64	Intel Emerald Rapids
GCP C4A	8	ARM64	Google Axion

Table 4.1: CPU details for different platforms

Results

Is reduced precision ray tracing enough?

Expected Results

- In single precision, we would expect to see
  - more lost particles
  - less reliable collision detections
  - potential numerical instability within a problem
  - some effect on the physical results
    - either local or global physical results

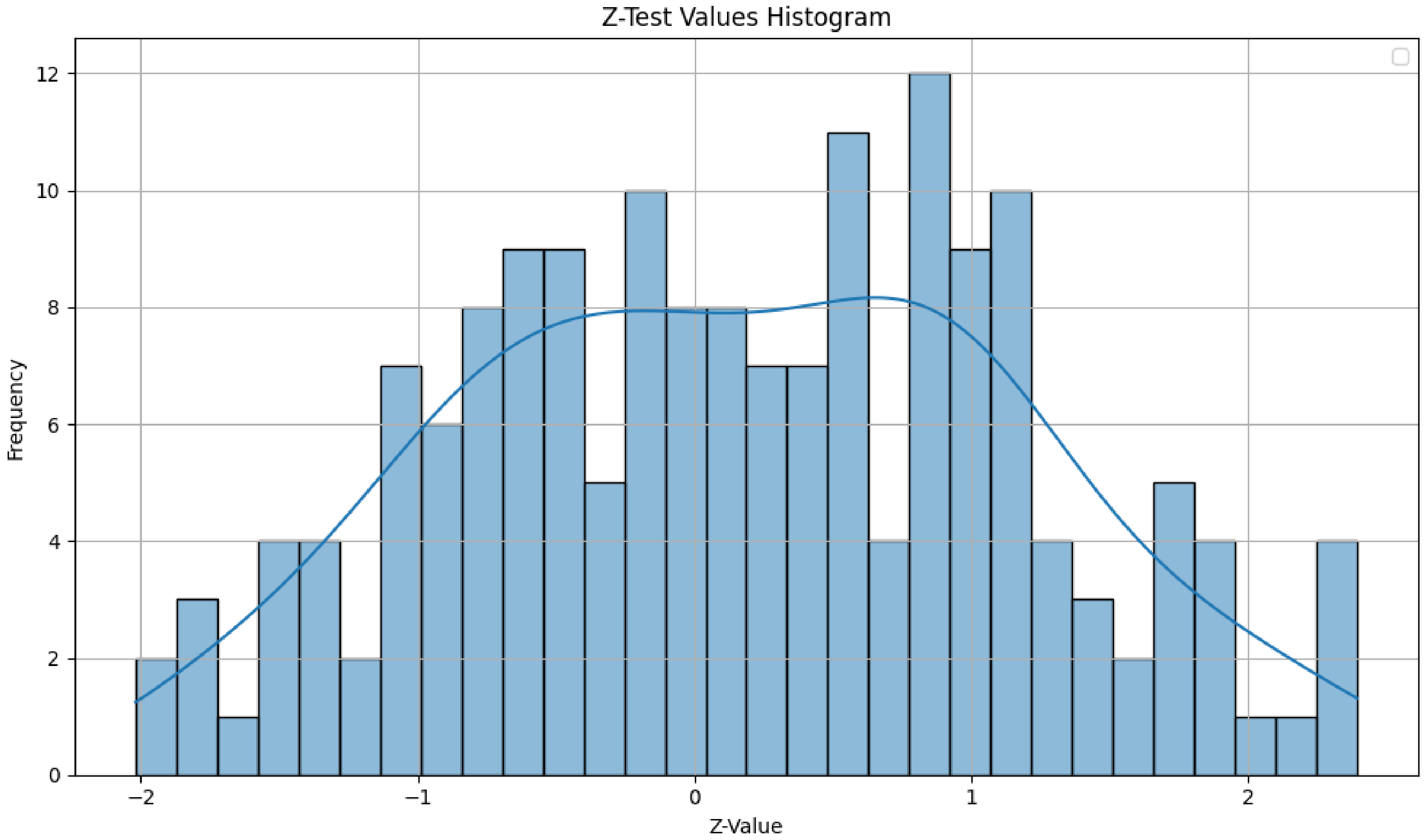
Although we can expect all of these to be true in theory, it is possible that the numerical impacts are so small that they would not even be reported by OpenMC.

What is a Good Result?

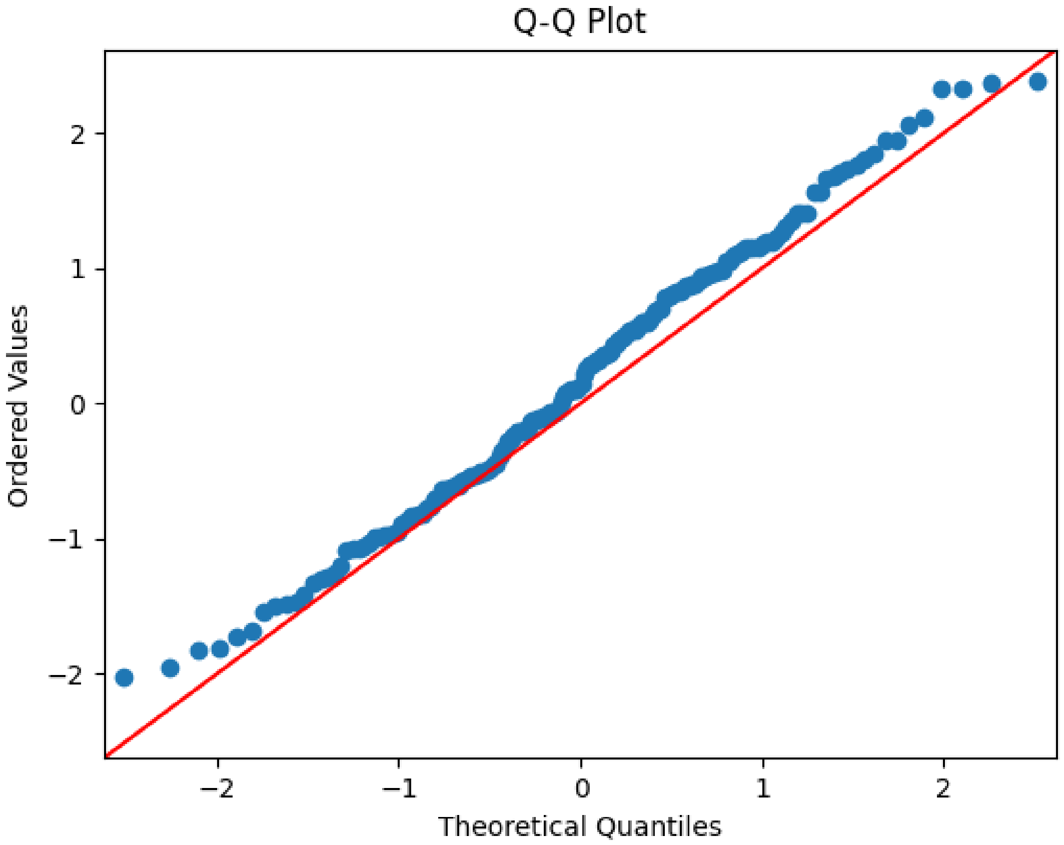
- In general, we define a good result as one where the results in single precision and double precision agree
  - This is formalized by applying a z-test to the results of a single precision simulation and a double precision simulation
- For a large set of problems to be considered in agreement, we would expect their z-test values to be normally distributed.

This definition of a good result is derived from the random behavior that OpenMC exhibits. This assumption is true for most MC codes.

Normalcy of Our Results



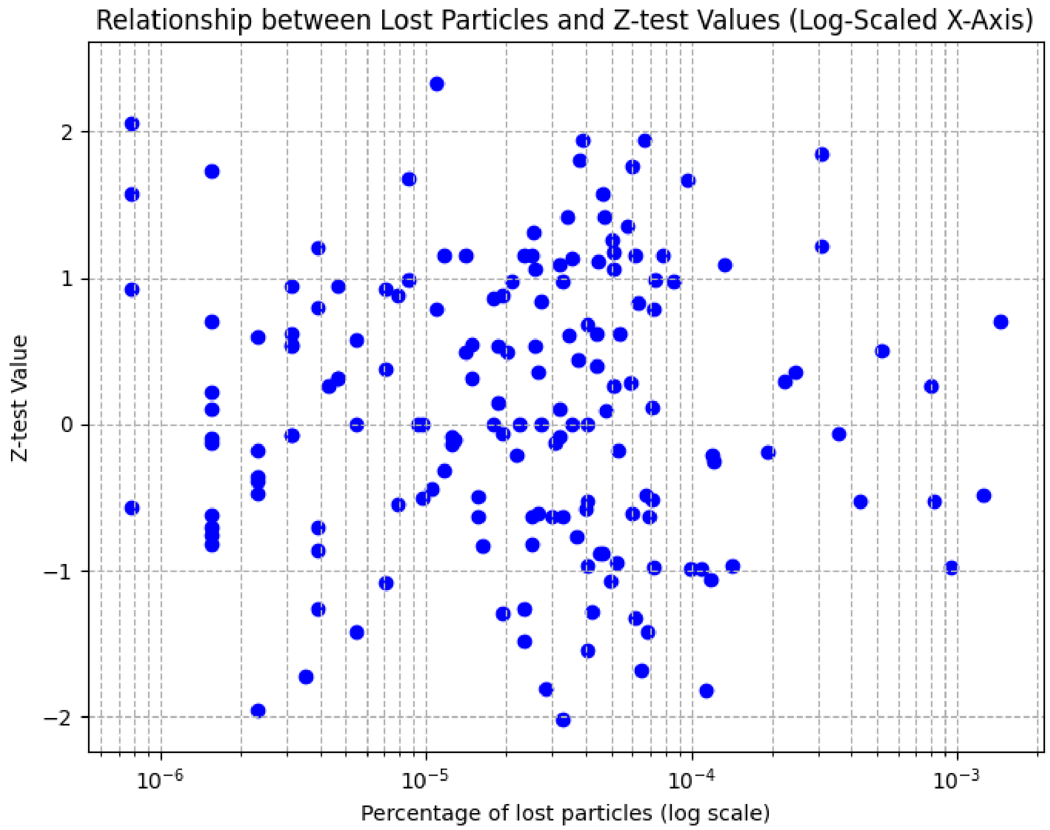
Normalcy of Our Results



Relationship to Lost Particles

- In several of our early discussions, researchers cited lost particles as their main area of concern regarding a switch to lower precision
- In double precision, the lost particle rate is astronomically low
- In single precision, the lost particle rate is much less predictable and mostly depends on the problem
  - In our simulations, we saw rates between 0.01% and over 95% of lost particles
  - We saw no relationship between lost particle rates and whether a problem agreed between its single precision and double precision simulations

Relationship to Lost Particles



Error Prone Problems

Table 5.1: Z-Test Results and Particle Loss for Reactor Problems

Problem Name	Z-Test Value	% Lost Particles
BWR	1,1419	95.5%
RBMK	207.06	2.75%
SFR	0.41451	0.000%
SMR	0.23570	0.00137%
TRIGA	4.1310	0.807%
VHTR	142.23	94.2%
VVER	160.23	66.3%

Performance Benefits of Single Precision

- Previous results have shown that a performance benefit can be gained by a switch to lower precision levels
  - Expected performance gains vary based on the study but agree that there is a benefit
  - These results are mostly constrained to a single platform or device
  - To test this, we took one problem (a small modular reactor) and tested its performance across a wide variety of platforms

In general, the performance of a set of operations done at a certain precision is related to the amount of silicon allocated to this precision level in the hardware.

Performance Benefits of Single Precision

Table 5.2: Speedup (Single vs. Double Precision) by Platform

Platform Name	**Simulation Speedup in Single-Precision**
MacBook Air	2.25x
GCP E2	1.02x
GCP C4	0.99x
GCP C4A	0.96x

Conclusions

What did we learn and what is next?

Takeaways and Next Steps

- Single precision and double precision produce indistinguishable results on simple problems, regardless of the rate of lost particles
  - This means that we cannot predict whether a problem is indistinguishable between precisions using only the lost particle rate
- We only studied a single global metric, so further study must be done to rule out the possibility of local biases
- Certain reactor models fail catastrophically with an insurmountable number of lost particles in single precision simulations
- Several simulations crashed in a single precision version of OpenMC
  - Further work needs to be done to develop a robust version in single precision

Questions?

Chris Marshall

chrismarshall@uchicago.edu