OPTIMISING A STOCHASTIC APPROACH TO SOLVING THE PARKER COSMIC RAY TRANSPORT EQUATION

A look at newer pseudo random number generators for use in stochastic simulations

Speaker: Phillip Strydom - Centre for Space Research, North-West University, Potchefstroom

Supervisors: Prof. Eugene Engelbrecht - Centre for Space Research, North-West University, Potchefstroom

Dr. Katlego Moloto - Centre for Space Research, North-West University, Potchefstroom

What are Cosmic Rays and what is the Parker Transport Equation

- Cosmic Rays (CR's)
  - Highly energetic particles
  - Multiple sources i.e. Jupiter or from outside of the solar system
  - Effected by the suns magnetic field and the plasma environment it travels through called the heliosphere
- Parker Transport Equation (PTE)
  - Mathematical model to describe the transport physics of CR's through the heliosphere
  - Includes effects of drifts, turbulence, diffusion

Solving the Parker Equation using a Stochastic Method

- Transform PTE into a set of Itō stochastic differential equations
- Use the Wiener Process to iteratively follow the particle step-by-step to the heliosphere boundary in a time-backwards manner
  - Use a pseudo random number generator to obtain a random time-step
  - Process requires Gaussian distributed values, Box-Muller Transform is used to meet this requirement
- Done for a large amount of particles and then averaged per energy step
- Model coded in Intel Fortran and parallelized using MPI
  - Previous research using this code was completed in part running on the CHPC Lengau cluster (see Moloto et al. 2018)

Optimizing the Stochastic Model Code and Testing Methodology

- Preliminary test of different PRNG’s to see if others have a faster runtime*
- Implement into existing transport code and run for different particle counts to get an idea of algorithm scalability*
- Compare runtimes to that of the currently used PRNG
- Compare output by looking at percentage deviation to confirm statistical viability

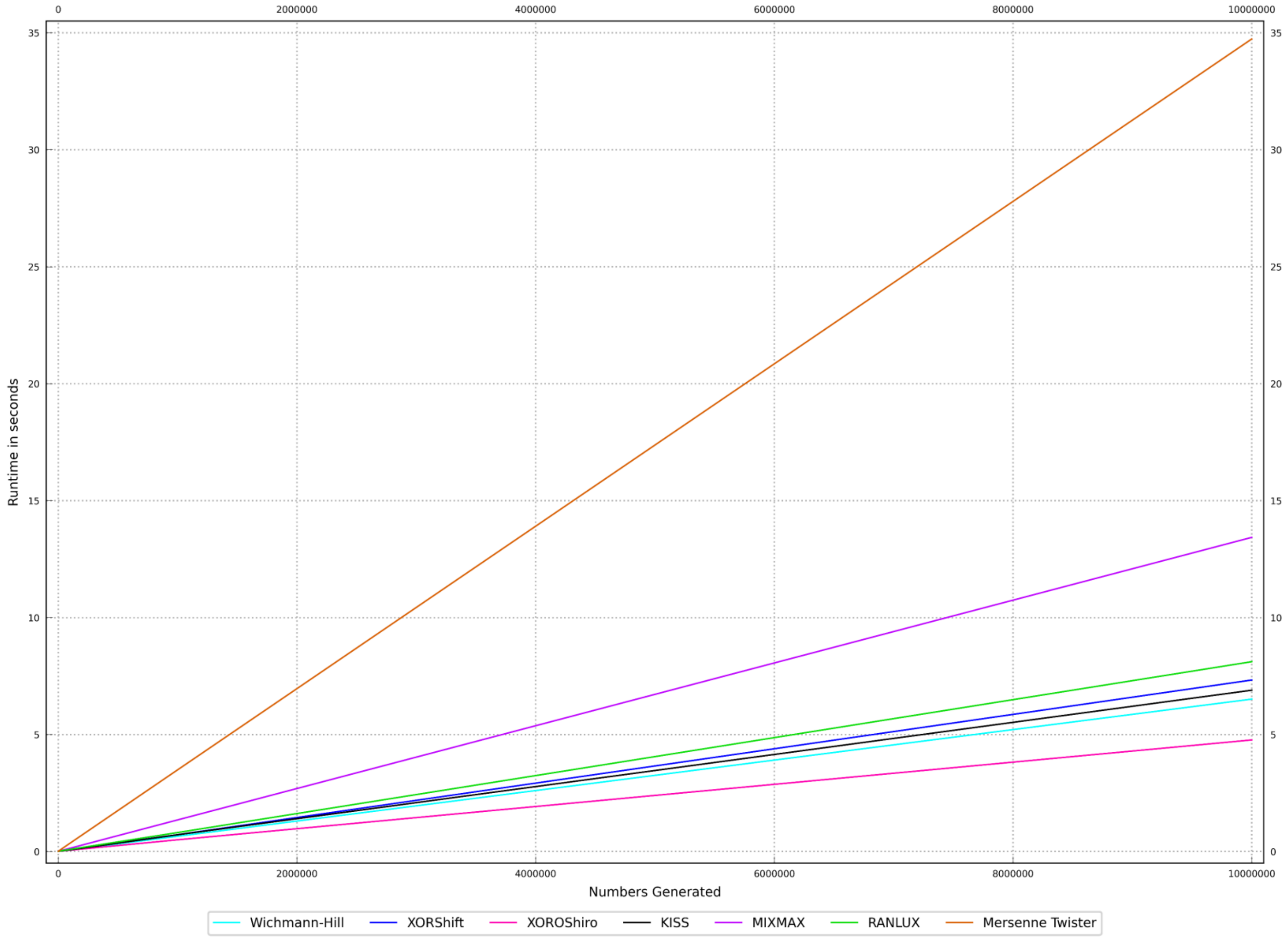
*All tests were performed on identical hardware using the Intel Fortran compiler with the code running parallel on 7 threads

Pseudo Random Number Generators Tested

- Current iteration of code implements the widely used Mersenne Twister
- Six other PRNG’s were identified*
  - Wichmann-Hill
  - RANLUX
  - KISS
  - XORShift
  - XOROShiro
  - MIXMAX

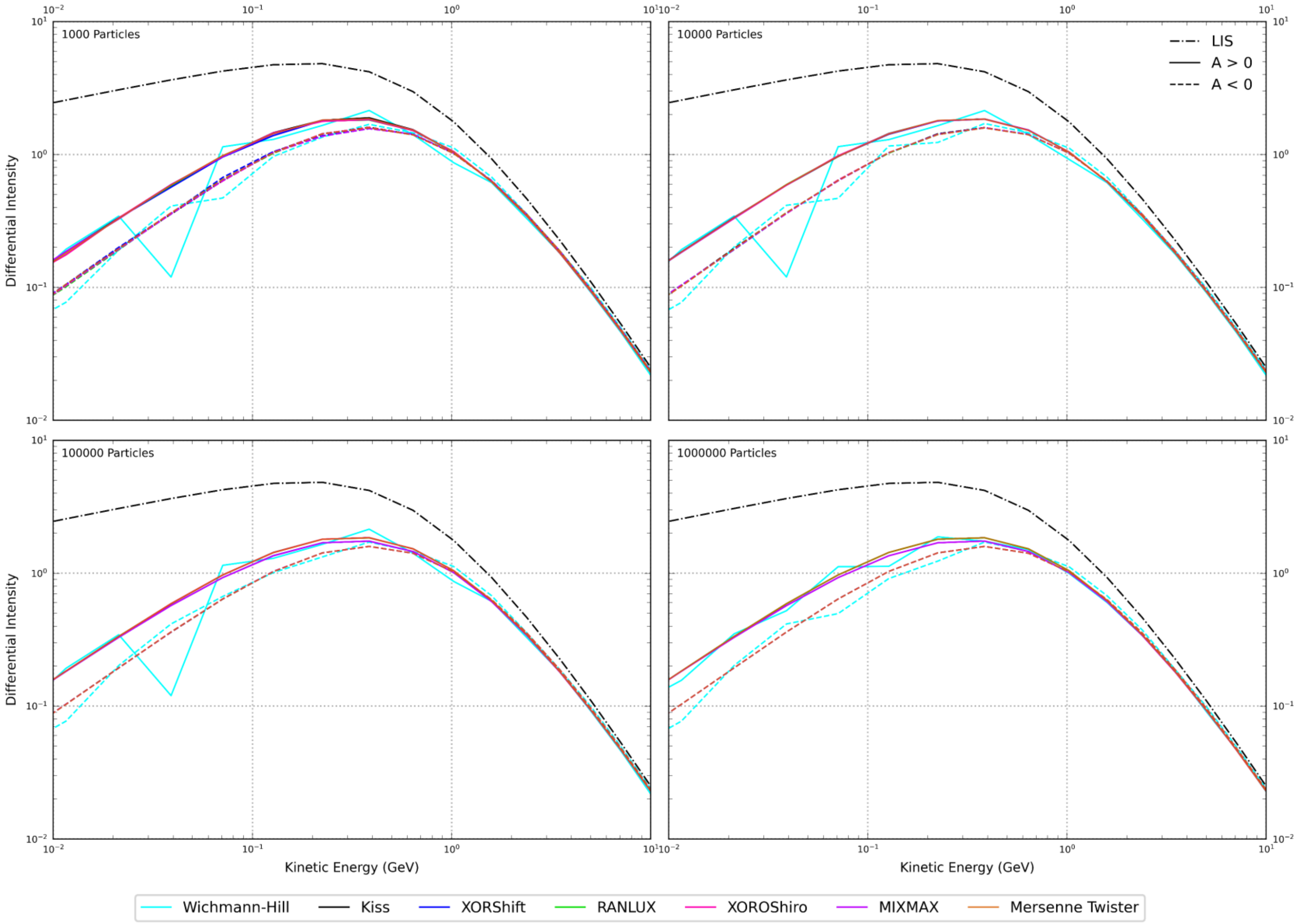
* Candidates were chosen on the basis that the code was freely licensed and publicly available in Fortran, other PRNG’s might be viable replacements

Results - Preliminary Runtime Tests



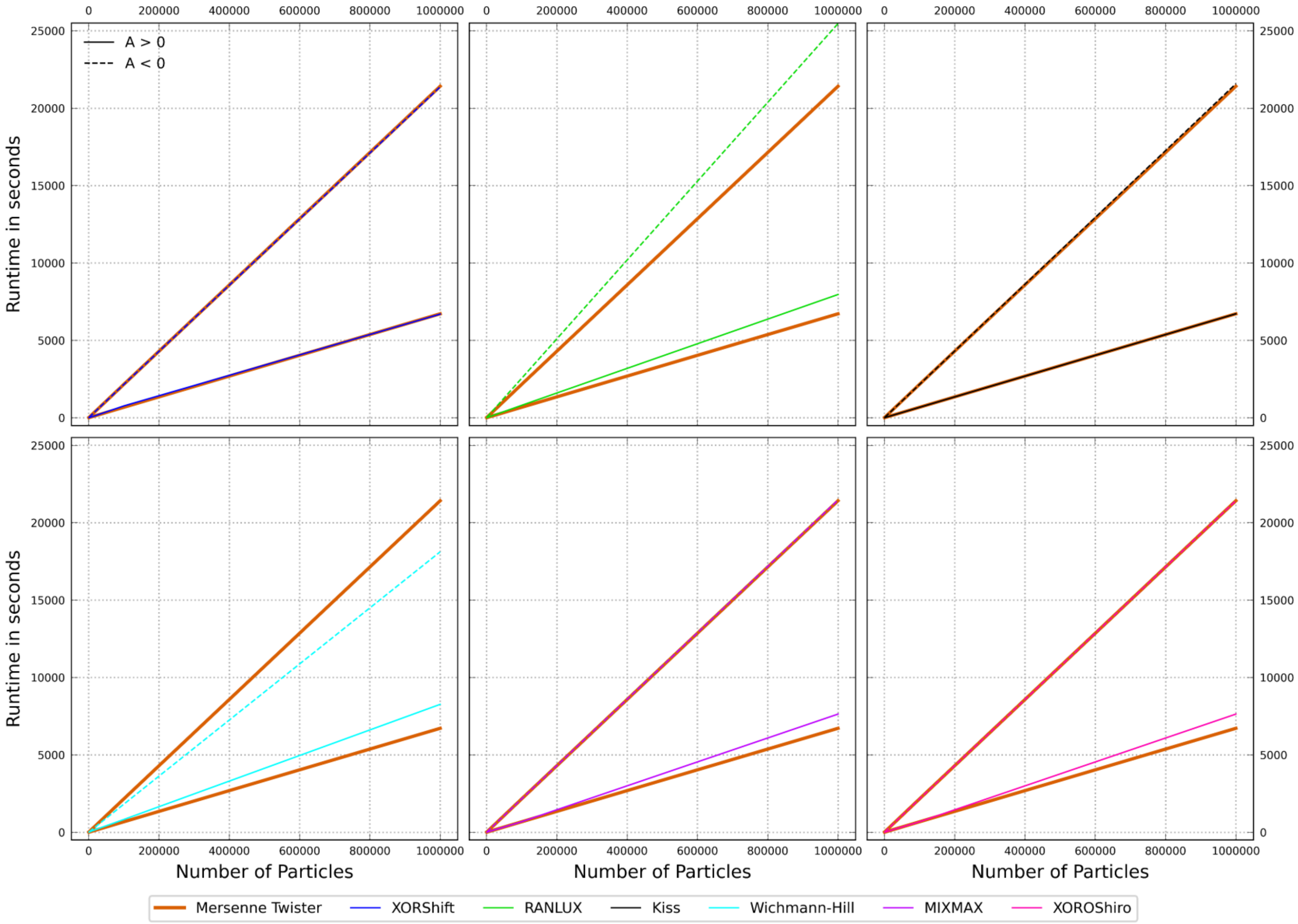
- All tested PRNG's seem to perform better than Mersenne Twister
- MIXMAX performed slower than expected
- XOROShiro is a clear standout

Results - Simulation Model Output



- All the PRNG's except for Wichmann-Hill seem to be accurate when compared to Mersenne Twister output
- MIXMAX and XOROShiro accuracy decrease as particle count increases

Results - Simulation Runtimes



- XORShift and KISS have the same runtime as that of Mersenne Twister
- RANLUX performed slower than expected
- MIXMAX and XOROShiro perform slower only for A>0 conditions

Conclusions

- Standouts are XORShift and KISS
  - Have the same or better statistical quality as Mersenne Twister with comparable runtimes
- Wichmann-Hill is clearly not viable
- XOROShiro and MIXMAX might be viable given further testing
- Further optimizations to the code itself might be the real source of performance improvements and should be considered
- Other transformation algorithms to replace Box-Muller, such as either the Ziggurat or Wallace methods should be considered and tested

Thank you