Towards a Brookhaven integrator for precision physics on GPUs

Robert Szafron

Peter Boyle, Sally Dawson, Taku Izubuchi, Tobias Neumann

Date 29 May 2022

FY2023 NPP LDRD Type A Proposal

Proposal title: Towards a Brookhaven integrator for precision physics on GPUs

Primary Investigator: Robert Szafron

Other Investigators: Peter Boyle, Sally Dawson, Taku Izubuchi. +Tobias Neumann (postdoc)

Indicate if this is a cross-directorate proposal. **No**

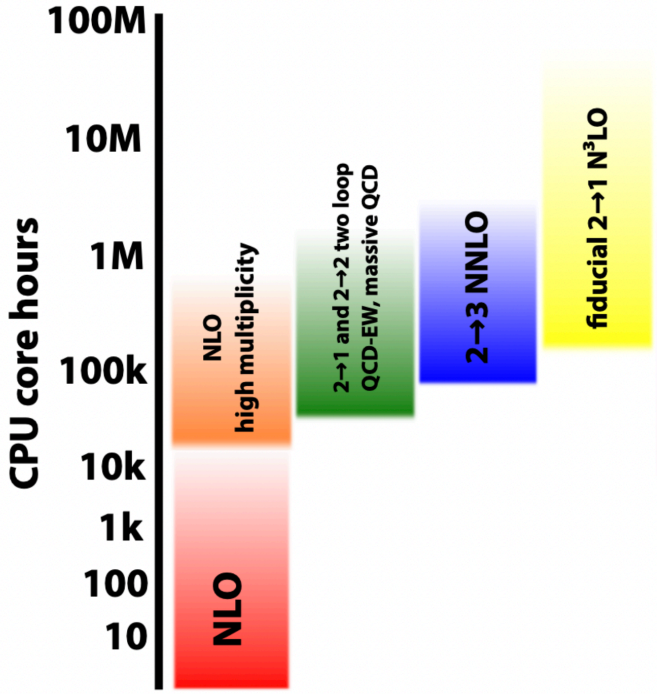
Program: Specify which program : **HEP**

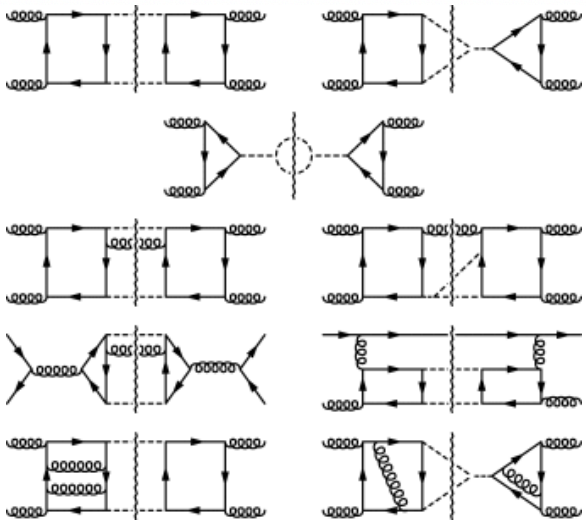
Proposal Term: From: 9/1/2022 To: 8/31/2025

Total funding per year in FY23, FY24 and FY25: 500k/500k/500k

Description of the LDRD Proposal

- Numerical integration of cross section phase space
- Computational cost / limited statistical precision is a barrier to higher order calculation, many particle final states.
- Need for better integrators than current approach (e.g. VEGAS)
- Propose a new direction based on algorithms used in statistical physics & lattice gauge theory to sample very high dimensional spaces.
- Enable higher order calculations to be more easily included in event generators, improving theory precision in current and future colliders.
- Opens opportunities for external funding: FOA and ECA (Szafron eligible)
  - **BNL/HET numerical integration software:**
  - new route to ASCR funding for HET

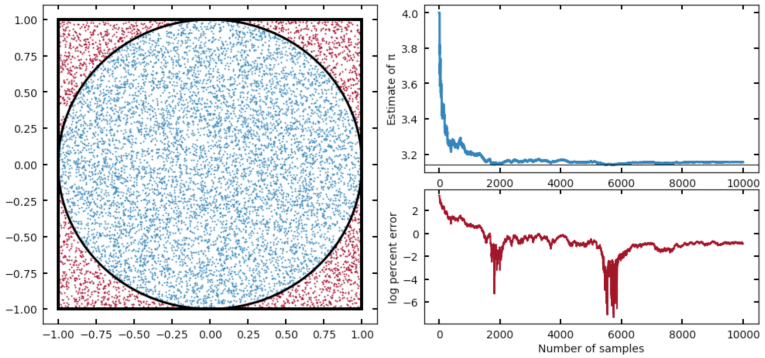




Monte Carlo integration uses random sampling to estimate an integral

coined by Nicholas Metropolis at LANL.

e.g. calculate π by throwing darts at a circle



Current approach to numerical integration uses importance sampling:

flat sampling: $\frac{1}{N}\sum_i f(x_i)$

importance sampling $\frac{1}{N}\sum_i \frac{f(x_i)}{P(x_i)}$

Current workhorse: VEGAS (Lepage)

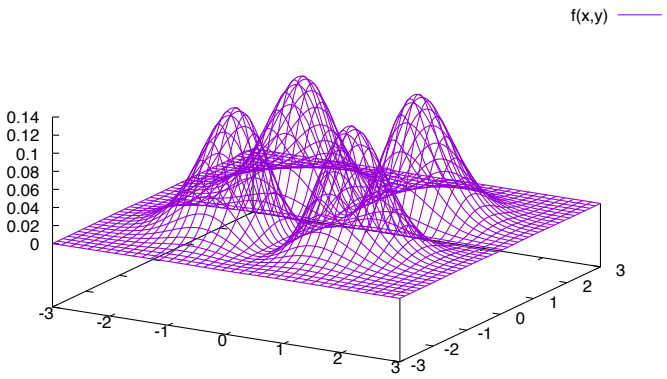
$$P(x,y,z) = X(x)\, Y(y)\, Z(z)$$

“Discovers” optimal X(x), Y(y), Z(z) by training on integrand

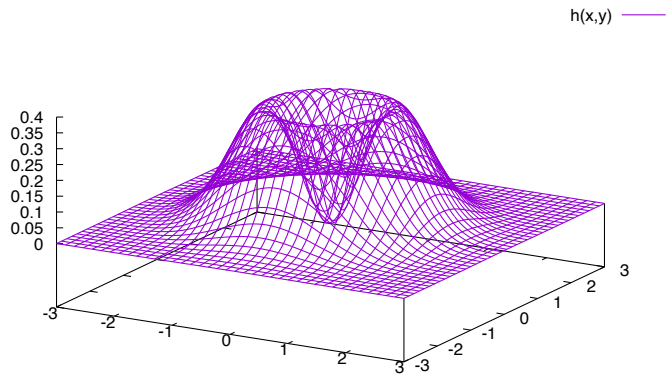
Factorizing probability distribution is easy to draw from

But cannot match general features so loses efficiency

Worse in high dimensions = higher order, more particles !



$$x^2 e^{-x^2} \times y^2 e^{-y^2}$$



$$(x^2 + y^2) e^{-(x^2+y^2)}$$

Likely solution: Markov Chain Monte Carlo

Can importance sample ANY distribution

Gives up simple drawing - samples by guided random walk – must decorrelate

A History of the Metropolis–Hastings Algorithm

David B. Hitchcock

The Metropolis–Hastings algorithm is an extremely popular Markov chain Monte Carlo technique among statisticians. This article explores the history of the algorithm, highlighting key personalities and events in its development. We relate reasons for the delay in the acceptance of the algorithm and reasons for its recent popularity.

KEY WORDS: Biography; Markov chain; Monte Carlo method; Simulation; Statistical computing.

1. INTRODUCTION

The Metropolis–Hastings (M–H) algorithm, a Markov chain Monte Carlo (MCMC) method, is one of the most popular techniques used by statisticians today. It is primarily used as a way to simulate observations from unwieldy distributions. The algorithm produces a Markov chain whose members' limiting distribution is the *target density* $\pi(x)$. At step j, an observation x_j is generated from an *instrumental density* $q(\cdot|x_i)$ (which is typically easy to simulate from). This candidate observation becomes the next value in the Markov chain with probability

$$\rho = \min\left\{\frac{\pi(x_j)q(x_i|x_j)}{\pi(x_i)q(x_j|x_i)}, 1\right\};$$

with probability $1-\rho$, set $x_j = x_i$, the previous value in the chain (Robert and Casella 1999, p. 233). Under certain conditions, the limiting distribution of the observations in the Markov chain is $\pi(x)$; see Chib and Greenberg (1995) for a detailed introduction.

Introduced in 1953, the M–H algorithm is a relatively old technique. Yet for decades it languished below the radar, outside the knowledge base of the typical statistician. Consider the statistical landscape just two decades ago. In the 1982 edition of the *Encyclopedia of Statistical Sciences*, there was no entry for "Metropolis–Hastings algorithm," "Metropolis," "Hastings," or "Markov chain Monte Carlo" (Kotz and Johnson 1982). Long after the method had originated and even after it had been theoretically validated, this comprehensive reference of all things statistical did not bother to mention it. What was the origin of the method and what factors accounted for its sudden rise to prominence?

2. THE EARLY DAYS OF MONTE CARLO METHODS

The use of Monte Carlo methods, defined broadly as the field of experiments using random numbers (Hammersley and Handscomb 1964, p. 2), existed well before the twentieth century. In 1777, Georges Louis Leclerc Comte de Buffon established a method for approximating π by repeatedly, randomly throwing a needle onto a grid of parallel lines and tracking how often the needle landed on a line (Liu 2001, p. vii). In the early twentieth century, William Gosset ("Student") used simulations with random numbers to help determine the sampling distributions of the correlation coefficient and the t statistic. But the mathematical branch of Monte Carlo methods really began in earnest in the 1940s among scientists at the Los Alamos Laboratory in New Mexico, which is where the seeds of the M–H algorithm were sown.

Nicholas C. Metropolis was born in 1915 in Chicago. He attended the University of Chicago, eventually receiving a doctorate in experimental physics there. He researched nuclear reactors with Enrico Fermi and Edward Teller, and, through his work with such noteworthy scientists, he came to the attention of J. Robert Oppenheimer, head of the Manhattan Project—the United States government's plan to build the first atomic bomb. In 1943, at the height of World War II, Oppenheimer recruited Metropolis to Los Alamos to develop mathematical equations to describe the states of physical materials (Ravo 1999). Annoyed by the slow, unwieldy electromechanical calculators they had to use, Metropolis and colleagues Richard Feynman and John von Neumann became interested in the prospect of fast electronic calculators (Santa Fe Institute Bulletin 2000).

After the war, Metropolis went back to the University of Chicago to teach, but in 1948 returned to Los Alamos, where state-sponsored research was burgeoning under America's top scientists. Metropolis led the design of the first programmable super-computer, which he called MANIAC (Mathematical Analyzer, Numerical Integrator and Computer) (Liu 2001, p. viii). Metropolis chose this name as a satirical poke at the acronyms favored by scientists, but it stuck.

Finally, the computing power was available to drive the development of Monte Carlo (MC) methods, and MC applications soon followed. The motivating example was the random behavior of neutrons in the fissile material in atomic bombs. Two leading mathematicians at Los Alamos, Stanislaw Ulam and John von Neumann, thought of the idea of performing computations via simulation, and Metropolis apparently coined the catchy name "Monte Carlo methods" (Liu 2001, p. viii). Motivated by their physics problems, Metropolis and Ulam (1949) introduced their idea to the statistics community in their article, "The Monte Carlo Method." They gave an example of estimating the probability of success of a solitaire strategy by undertaking the strategy in many trials and tracking what proportion were successful. Appealing to the theory of probability, they noted, "The estimate will never be confined within given limits with

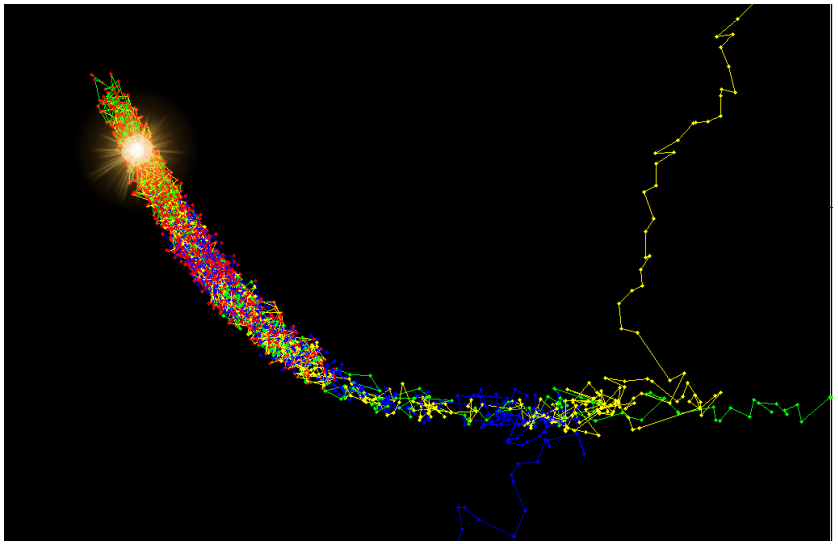
History [edit]

The algorithm was named after Nicholas Metropolis, who authored the 1953 article *Equation of State Calculations by Fast Computing Machines* together with Arianna W. Rosenbluth, Marshall Rosenbluth, Augusta H. Teller and Edward Teller. This article proposed the algorithm for the case of symmetrical proposal distributions, and W. K. Hastings extended it to the more general case in 1970.[1]

Some controversy exists with regard to credit for development of the algorithm. Metropolis, who was familiar with the computational aspects of the method, had coined the term "Monte Carlo" in an earlier article with Stanisław Ulam, and led the group in the Theoretical Division that designed and built the MANIAC I computer used in the experiments in 1952. However, prior to 2003, there was no detailed account of the algorithm's development. Then, shortly before his death, Marshall Rosenbluth attended a 2003 conference at LANL marking the 50th anniversary of the 1953 publication. At this conference, Rosenbluth described the algorithm and its development in a presentation titled "Genesis of the Monte Carlo Algorithm for Statistical Mechanics".[2] Further historical clarification is made by Gubernatis in a 2005 journal article[3] recounting the 50th anniversary conference. Rosenbluth makes it clear that he and his wife Arianna did the work, and that Metropolis played no role in the development other than providing computer time.

This contradicts an account by Edward Teller, who states in his memoirs that the five authors of the 1953 article worked together for "days (and nights)".[4] In contrast, the detailed account by Rosenbluth credits Teller with a crucial but early suggestion to "take advantage of statistical mechanics and take ensemble averages instead of following detailed kinematics". This, says Rosenbluth, started him thinking about the generalized Monte Carlo approach – a topic which he says he had discussed often with John Von Neumann. Arianna Rosenbluth recounted (to Gubernatis in 2003) that Augusta Teller started the computer work, but that Arianna herself took it over and wrote the code from scratch. In an oral history recorded shortly before his death,[5] Rosenbluth again credits Teller with posing the original problem, himself with solving it, and Arianna with programming the computer. In terms of reputation, there is little reason to question Rosenbluth's account. In a biographical memoir of Rosenbluth, Freeman Dyson writes:[6]

> Many times I came to Rosenbluth, asking him a question [...] and receiving an answer in two minutes. Then it would usually take me a week of hard work to understand in detail why Rosenbluth's answer was right. He had an amazing ability to see through a complicated physical situation and reach the right answer by physical arguments. Enrico Fermi was the only other physicist I have known who was equal to Rosenbluth in his intuitive grasp of physics.



Summary

Combine expertise from precision and lattice in HET group

Opportunity to improve precision of numerical integration

Feed into event generation for colliders – HEP area impact

Need emerges from Computational Frontier Snowmass working group: HL-LHC and EIC.

ECA and HEP Advanced Computing FOA funding opportunities

Develop BNL software capability & ASCR funding case

In statistics and statistical physics, the **Metropolis–Hastings algorithm** is a Markov chain Monte Carlo (MCMC) method for obtaining a sequence of random samples from a probability distribution from which direct sampling is difficult. This sequence can be used to approximate the distribution (e.g. to generate a histogram) or to compute an integral (e.g. an expected value). Metropolis–Hastings and other MCMC algorithms are generally used for sampling from multi-dimensional distributions, especially when the number of dimensions is high. For single-dimensional distributions, there are usually other methods (e.g. adaptive rejection sampling) that can directly return independent samples from the distribution, and these are free from the problem of autocorrelated samples that is inherent in MCMC methods.

- VEGAS style methods running out of gas at 3,4,5 loop – 12-20 dofs
- MCMC used in lattice up to billions of dofs in guided random walk
  - MCMC used in VEGAS+ as a `preconditioner' to set the sampling
- Use to sample the momenta in integrals in precision physics
- Potential to offload to GPU's; combine LQCD expertise with precision expertise
- Quite possibly better than recent ML/AI based research
  - Detailed comparison to VEGAS+ as function of dimensionality and integrand

FY22 Initiative : HEP Advanced Computing

- The Office of Science aims to develop the computing infrastructure necessary to support future scientific discoveries by identifying computing needs common across programs and the infrastructure necessary to support them
- The Office of High Energy Physics will support the SC Advanced Computing Initiative with new activities across four HEP Programs
    - Energy Frontier – work to develop expanded HPC usage for LHC experiments. This work continues to push high throughput computing workflows onto DOE SC HPCs.
    - Intensity Frontier – work to develop HPC applications for DUNE, the Short Baseline Neutrino program, and Liquid Argon TPCs using high throughput computing workflows.
    - General Accelerator R&D – work to develop scheduling across DOE SC User Facilities to coordinate experiment data taking and HPC computing time to process experimental data in real-time using HPCs.
    - Theory – work to expand use of HPCs for high order correction calculations for perturbative QCD. These complex simulations are needed to both produce simulations of sufficiently fidelity to compare with data and address some of the High Luminosity LHC computing simulation needs.
- These new activities have significant relevance to work in other SC offices
    - Common interest with ASCR, BES, FES, and NP