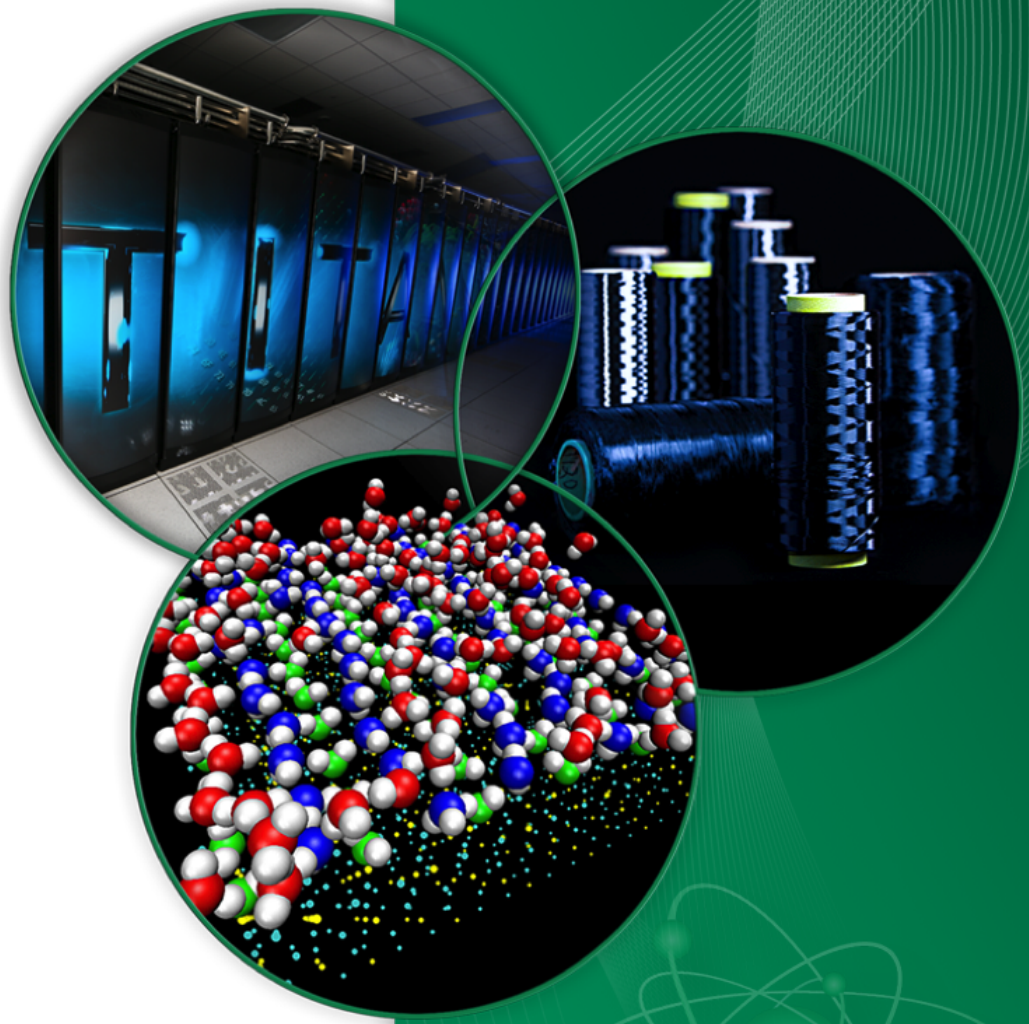
Monte Carlo Linear Solvers



Stuart Slattery
Tom Evans
Steven Hamilton

Oak Ridge National Laboratory

November 4, 2014

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research program.

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

OAK RIDGE National Laboratory

Motivation

- As we move towards exascale computing, the rate of errors is expected to increase dramatically
  - The probability that a compute node will fail during the course of a large scale calculation may be near 1
- Algorithms need to not only have increased concurrency/scalability but have the ability to recover from hardware faults
  - Lightweight machines
  - Heterogeneous machines
  - Both characterized by low power and high concurrency

OAK RIDGE National Laboratory

Towards Exascale Concurrency and Resiliency

- Two basic strategies:

1. State with current "state of the art" methods and make incremental modifications to improve scalability and fault tolerance
   - Many efforts are heading in this direction, attempting to find additional concurrency to exploit

2. Start with methods having natural scalability and resiliency aspects and work at improving performance (e.g. Monte Carlo)
   - Soft failures buried within the tally variance
   - Hard failures mitigated by replication
   - Concurrency enabled by several levels of parallelism

OAK RIDGE National Laboratory

Monte Carlo Methods

Monte Carlo for Linear Systems

- Suppose we want to solve $\mathbf{Ax} = \mathbf{b}$
- If $\rho(\mathbf{I} - \mathbf{A}) < 1$, we can write the solution using the Neumann series

$$\mathbf{x} = \sum_{n=0}^{\infty} (\mathbf{I} - \mathbf{A})^n \mathbf{b} = \sum_{n=0}^{\infty} \mathbf{H}^n \mathbf{b}$$

  where $\mathbf{H} \equiv (\mathbf{I} - \mathbf{A})$ is the Richardson iteration matrix

- Build the Neumann series stochastically

$$x_i = \sum_{k=0}^{\infty} \sum_{i_1}^{N} \sum_{i_2}^{N} \cdots \sum_{i_k}^{N} h_{i,i_1} h_{i_1,i_2} \ldots h_{i_{k-1},i_k} b_{i_k}$$

- Define a sequence of state transitions

$$\nu = i \to i_1 \to \cdots \to i_{k-1} \to i_k$$

OAK RIDGE National Laboratory

Forward Monte Carlo

- Choose a row-stochastic matrix $\mathbf{P}$ and weight matrix $\mathbf{W}$ such that $\mathbf{H} = \mathbf{P} \circ \mathbf{W}$
- Typical choice (Monte Carlo Almost-Optimal):

$$\mathbf{P}_{ij} = \frac{|\mathbf{H}_{ij}|}{\sum_{j=1}^{N} |\mathbf{H}_{ij}|}$$

- To compute solution component $\mathbf{x}_i$:
  - Start a history in state i (with initial weight of 1)
  - Transition to new state j based probabilities determined by $\mathbf{P}_i$
  - Modify history weight based on corresponding entry in $\mathbf{W}_{ij}$
  - Add contribution to $\mathbf{x}_i$ based on current history weight and value of $\mathbf{b}_j$
- A given random walk can only contribute to a single component of the solution vector

OAK RIDGE National Laboratory

Sampling Example (Forward Monte Carlo)

- Suppose

$$\mathbf{A} = \begin{bmatrix} 1.0 & -0.2 & -0.6 \\ -0.4 & 1.0 & -0.4 \\ -0.1 & -0.4 & 1.0 \end{bmatrix} \rightarrow \mathbf{H} \equiv (\mathbf{I} - \mathbf{A}) = \begin{bmatrix} 0.0 & 0.2 & 0.6 \\ 0.4 & 0.0 & 0.4 \\ 0.1 & 0.4 & 0.0 \end{bmatrix}$$

  then

$$\mathbf{P} = \begin{bmatrix} 0.0 & 0.25 & 0.75 \\ 0.5 & 0.0 & 0.5 \\ 0.2 & 0.8 & 0.0 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} 0.0 & 0.8 & 0.8 \\ 0.8 & 0.0 & 0.8 \\ 0.5 & 0.5 & 0.0 \end{bmatrix}$$

- If a history is started in state 3, there is a 20% chance of it transitioning to state 1 and an 80% chance of moving to state 2

OAK RIDGE National Laboratory

Solving the Heat Equation: Forward Method

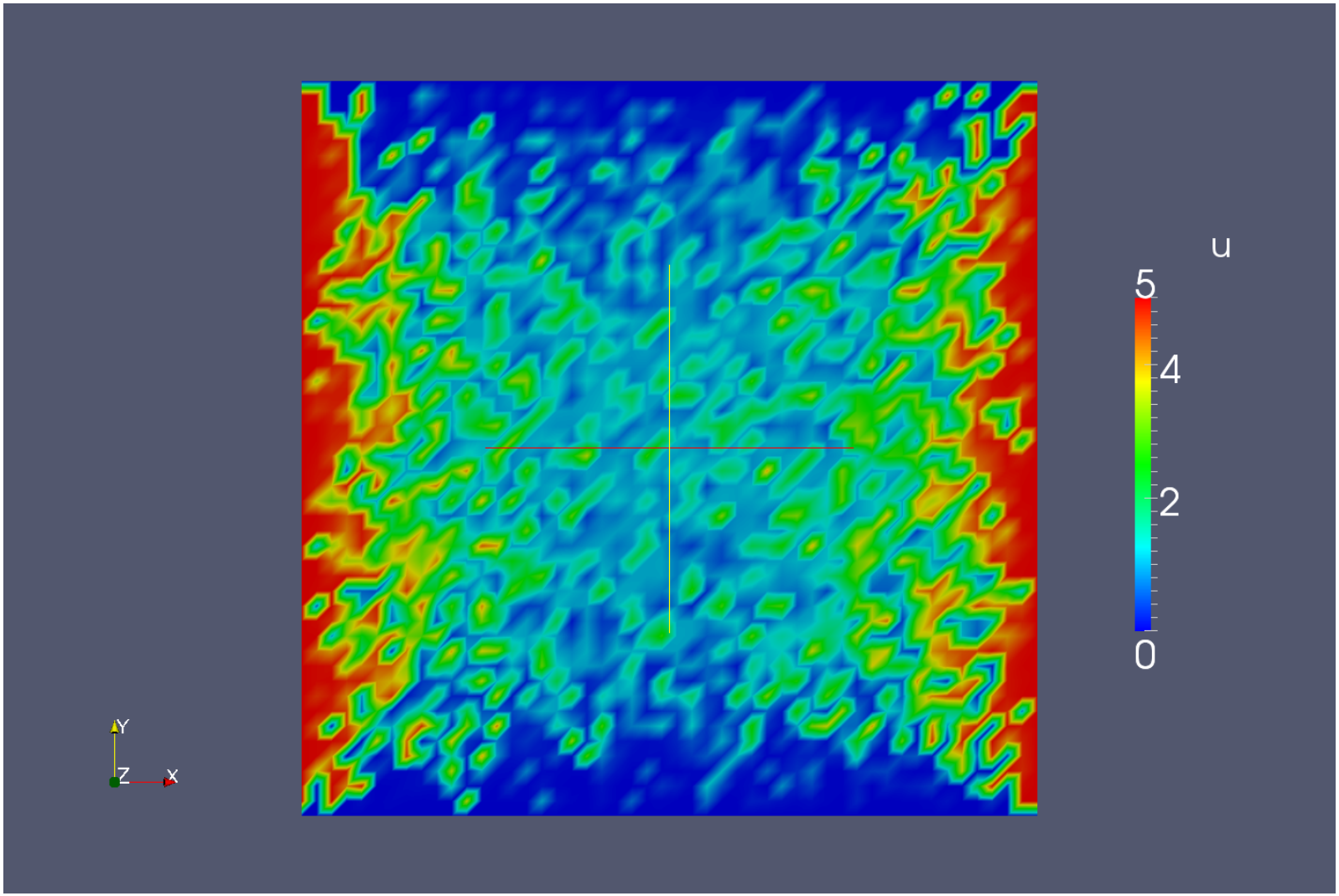


Figure : **Forward solution.** 2.5×10^3 *total histories.*

OAK RIDGE National Laboratory

Solving the Heat Equation: Forward Method

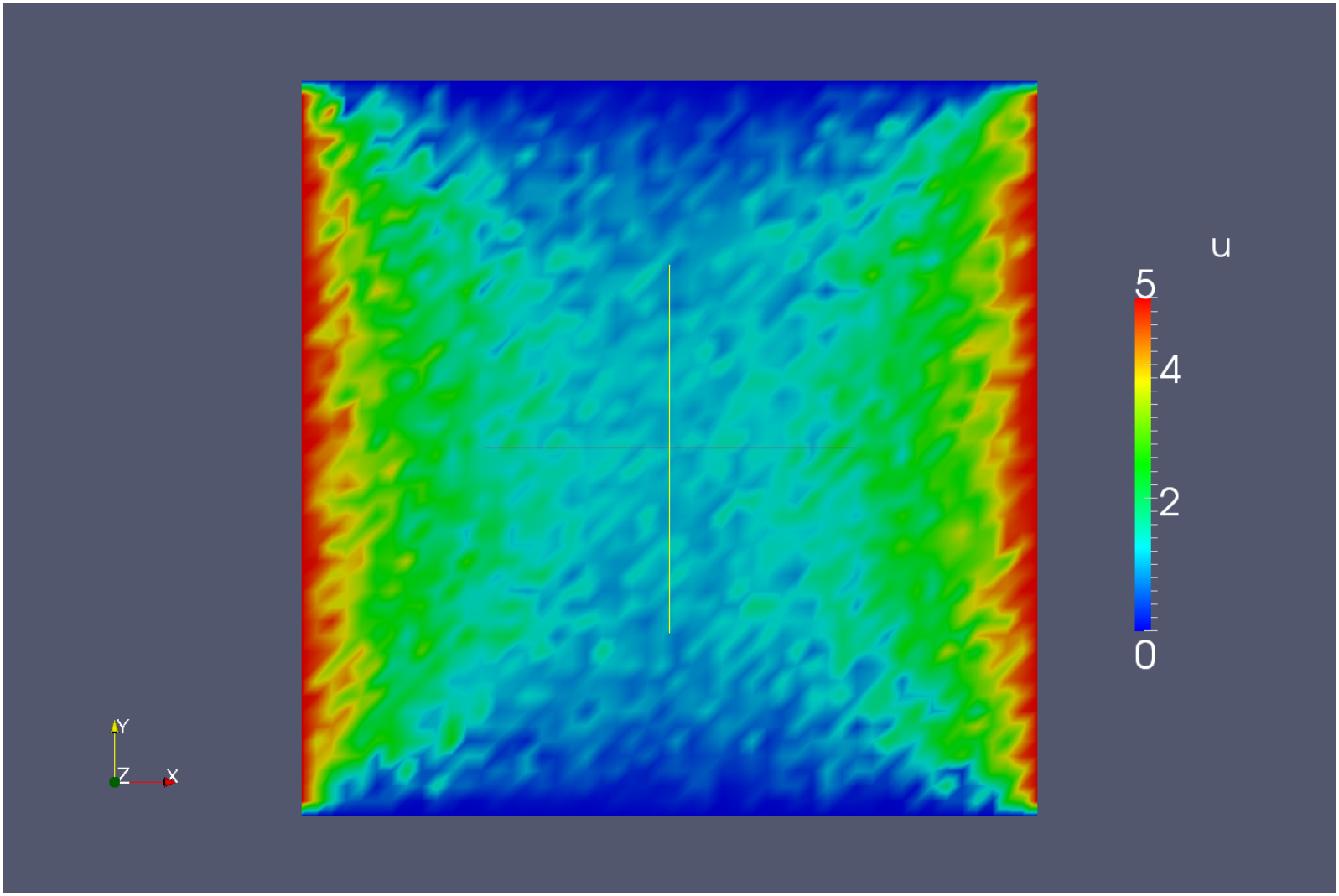


Figure : **Forward solution.** 2.5×10^4 *total histories.*

OAK RIDGE National Laboratory

Solving the Heat Equation: Forward Method

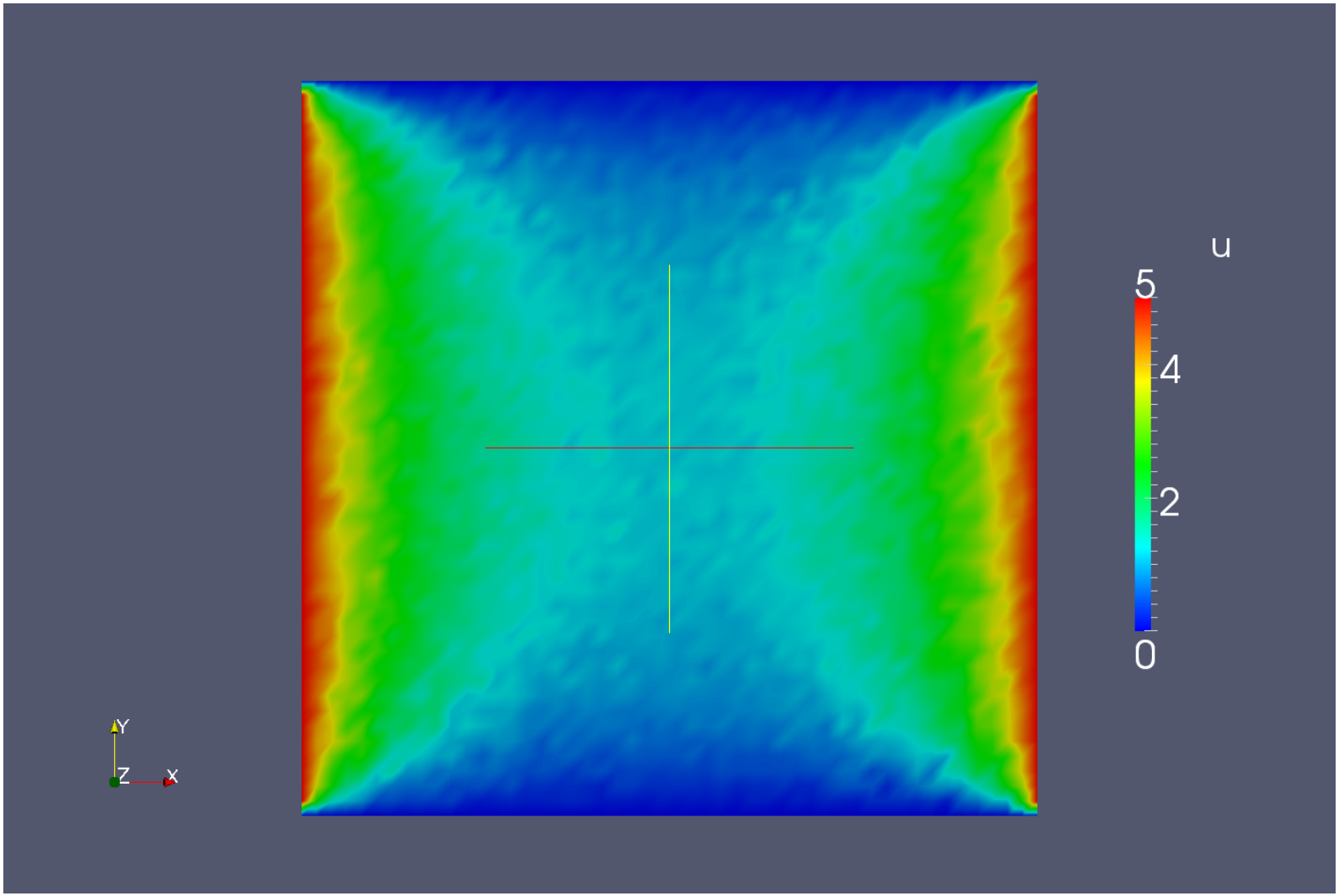


Figure : **Forward solution.** 2.5×10^5 *total histories.*

OAK RIDGE National Laboratory

Solving the Heat Equation: Forward Method

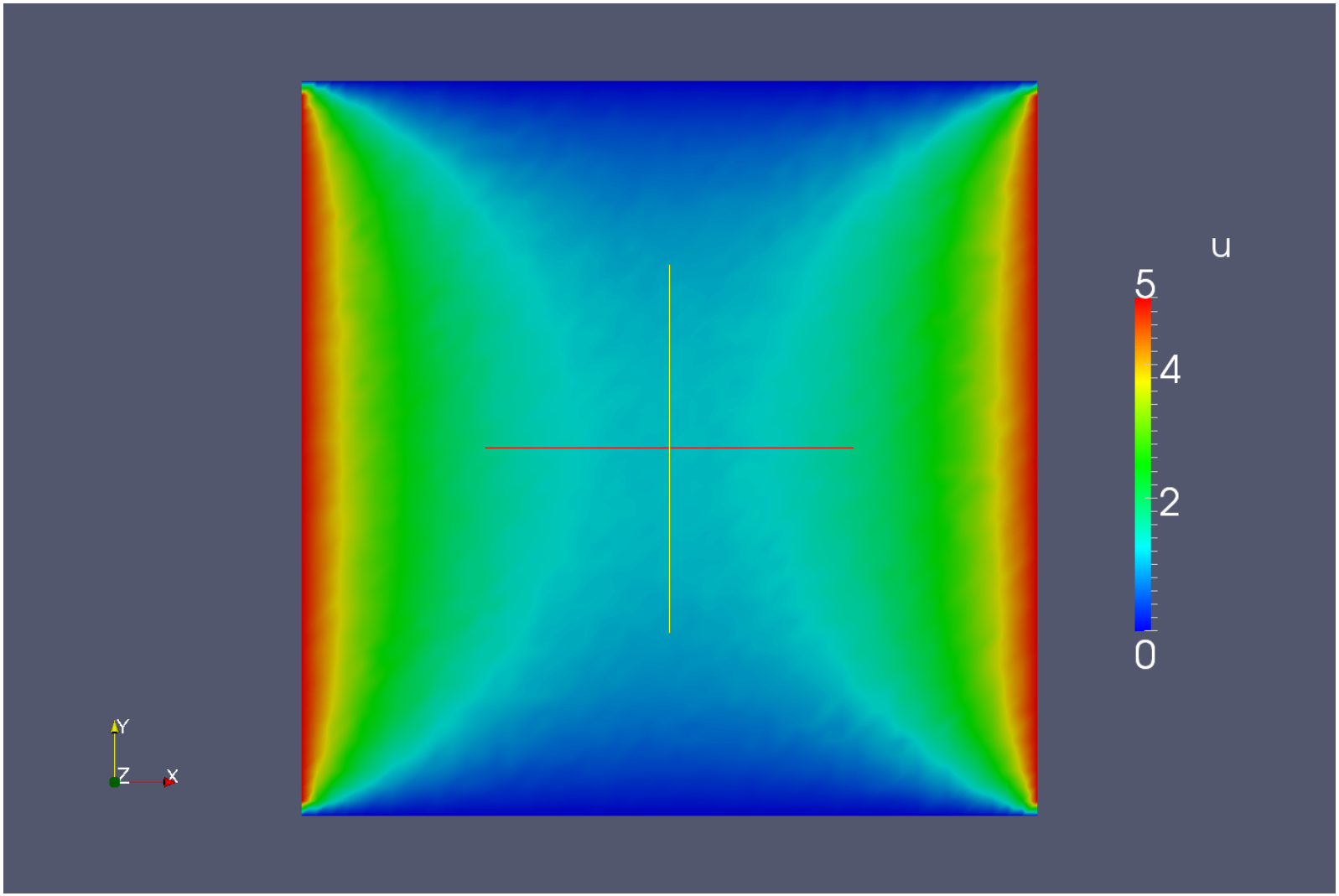


Figure : **Forward solution.** 2.5×10^6 *total histories.*

OAK RIDGE National Laboratory

Adjoint Monte Carlo

- Choose $\mathbf{P}$ and $\mathbf{W}$ such that $\mathbf{H}^T = \mathbf{P} \circ \mathbf{W}$
- Typical choice (Monte Carlo Almost-Optimal):

$$\mathbf{P}_{ij} = \frac{|\mathbf{H}_{ji}|}{\sum_{i=1}^{N} |\mathbf{H}_{ji}|}$$

- To estimate solution:
  - Start a history in random state i by sampling from distribution determined by $\mathbf{b}$
  - Transition to new state j based probabilities determined by $\mathbf{P}_i$
  - Modify history weight based on corresponding entry in $\mathbf{W}_{ij}$
  - Add contribution to $\mathbf{x}_j$ based on current history weight
- A given random walk can contribute to many different components of the solution vector

OAK RIDGE National Laboratory

Solving the Heat Equation: Adjoint Method

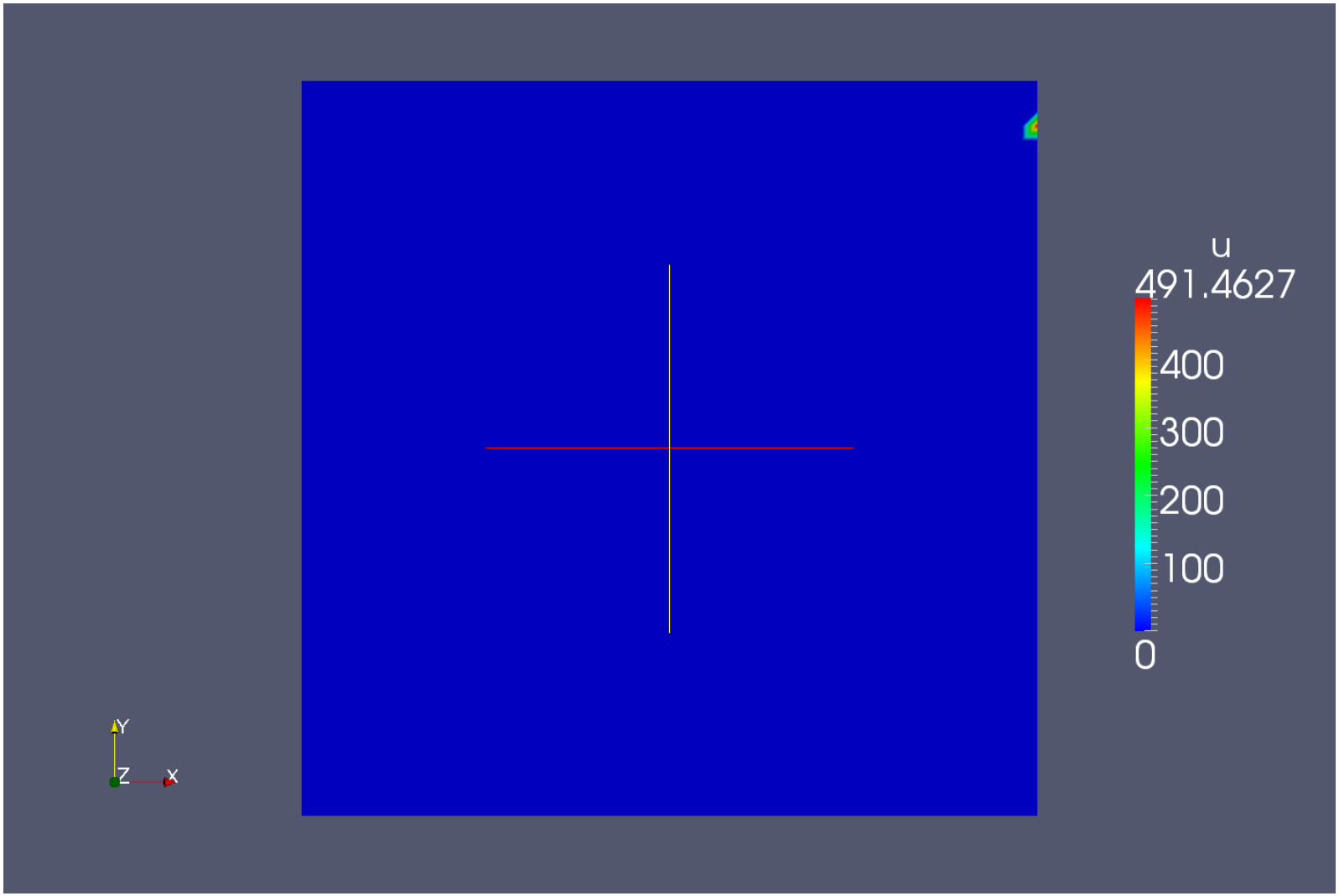


Figure : **Adjoint solution.** 1×10^0 *total histories.*

OAK RIDGE National Laboratory

Solving the Heat Equation: Adjoint Method

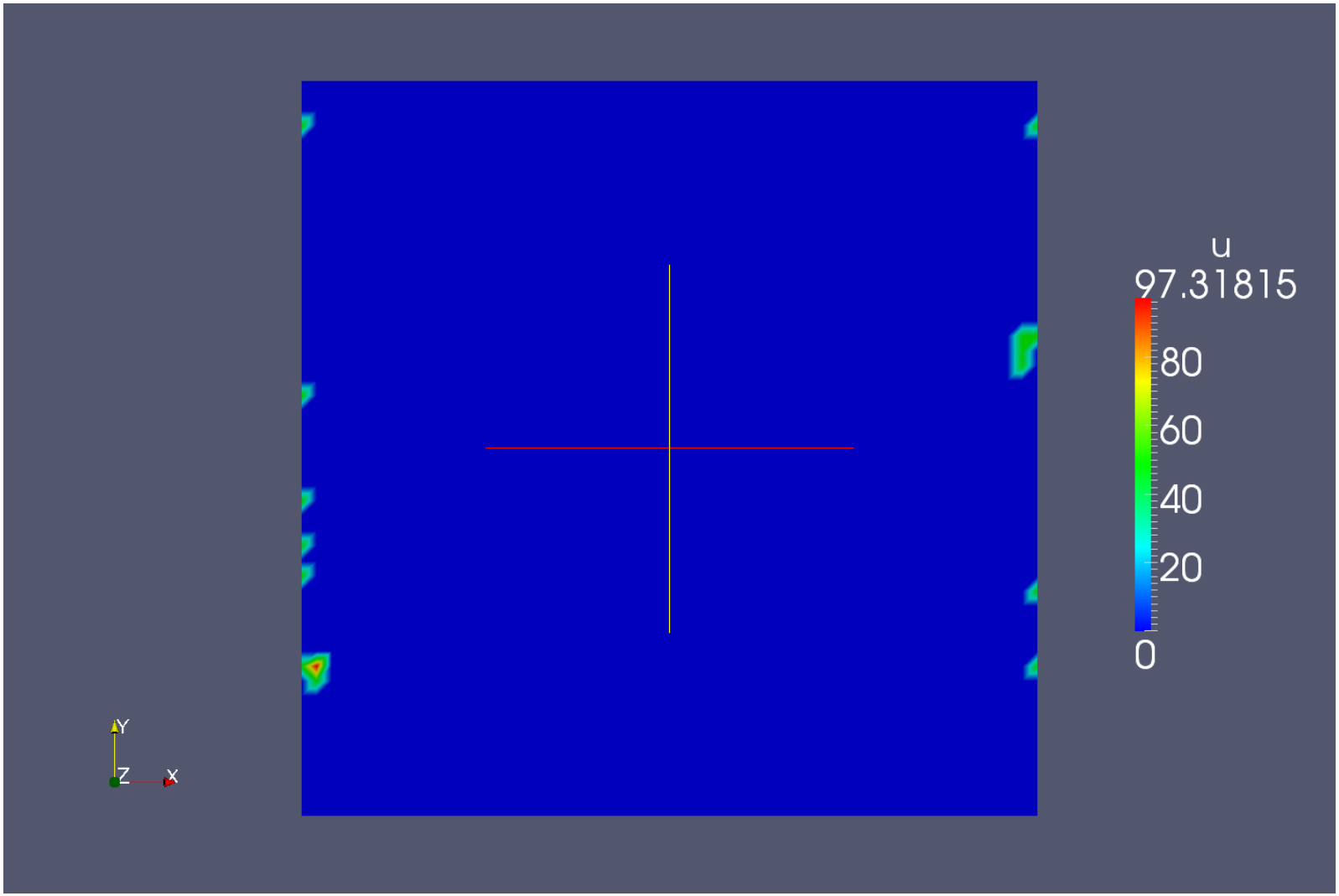


Figure : **Adjoint solution.** 1×10^1 *total histories.*

OAK RIDGE National Laboratory

Solving the Heat Equation: Adjoint Method

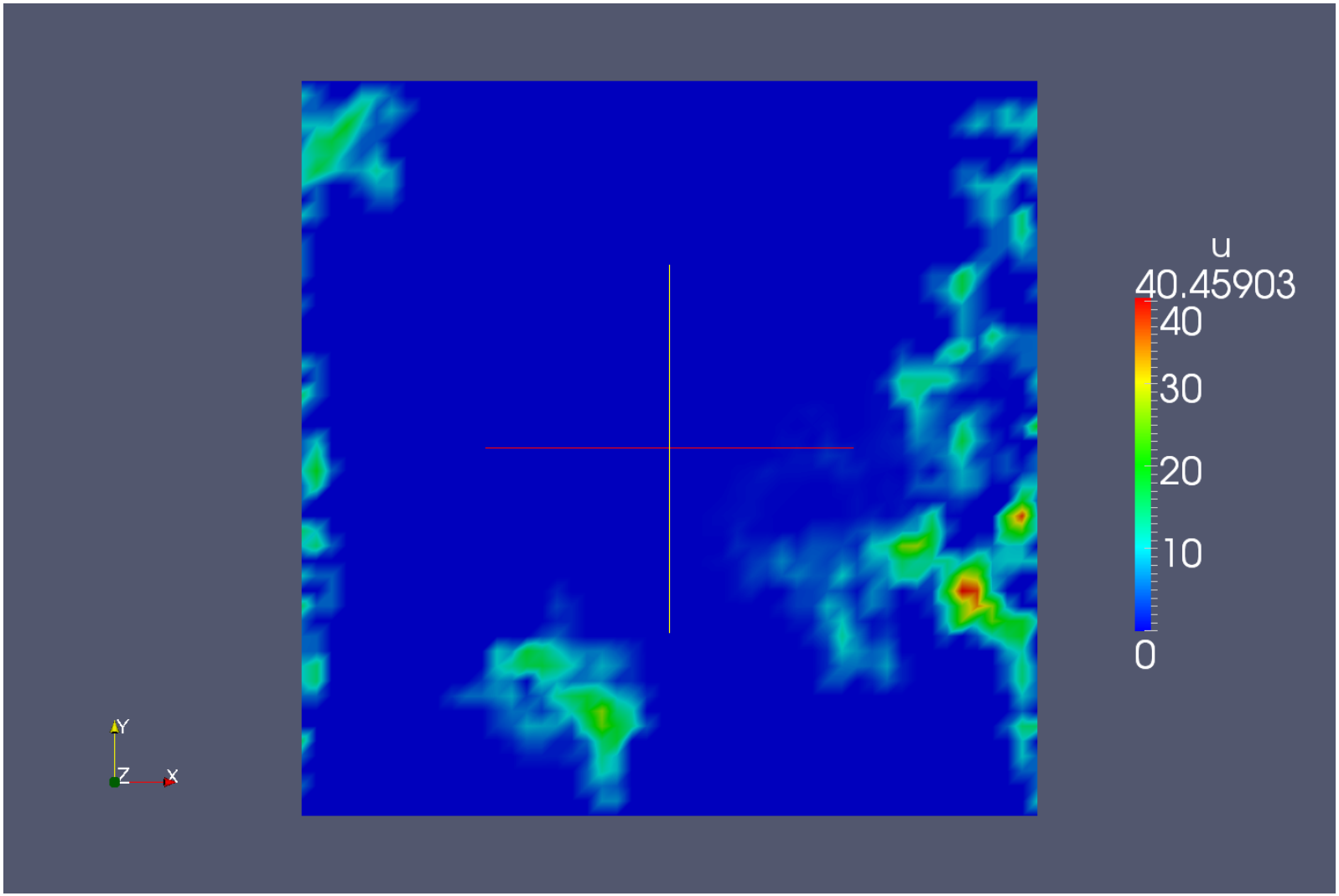


Figure : **Adjoint solution.** 1×10^2 *total histories.*

OAK RIDGE National Laboratory

Solving the Heat Equation: Adjoint Method

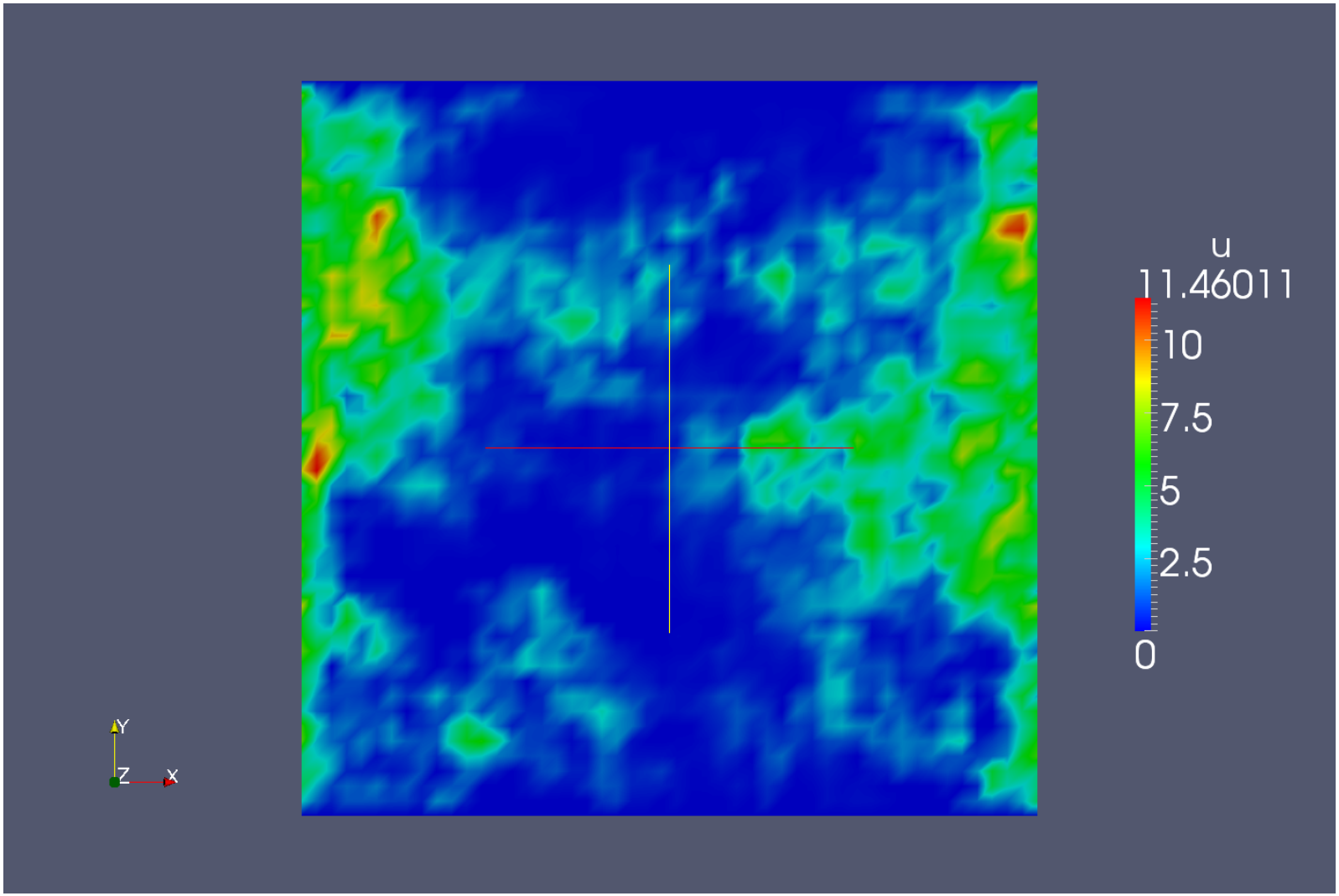


Figure : **Adjoint solution.** 1×10^3 *total histories.*

OAK RIDGE National Laboratory

Solving the Heat Equation: Adjoint Method

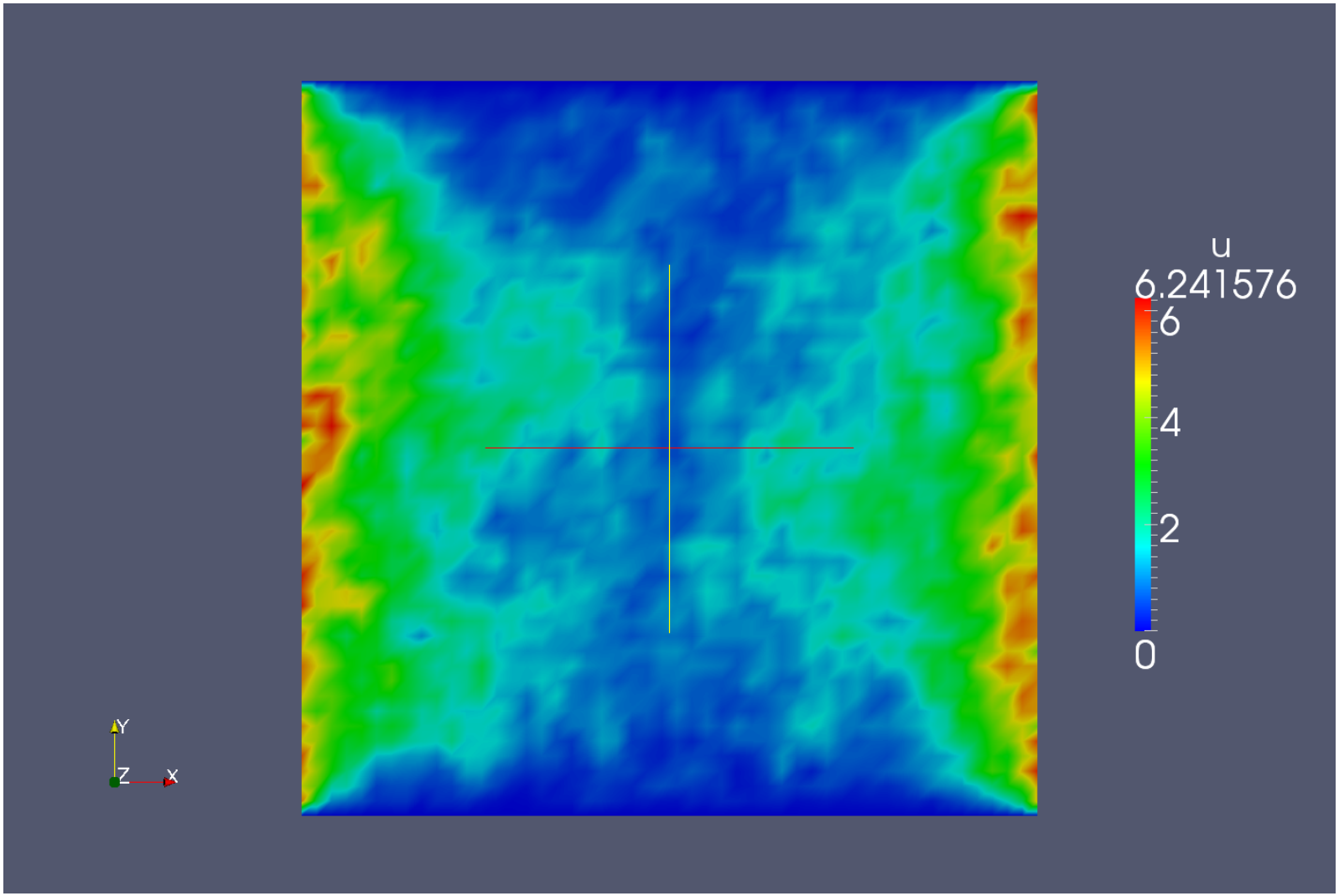


Figure : **Adjoint solution.** 1×10^4 *total histories.*

OAK RIDGE National Laboratory

Solving the Heat Equation: Adjoint Method

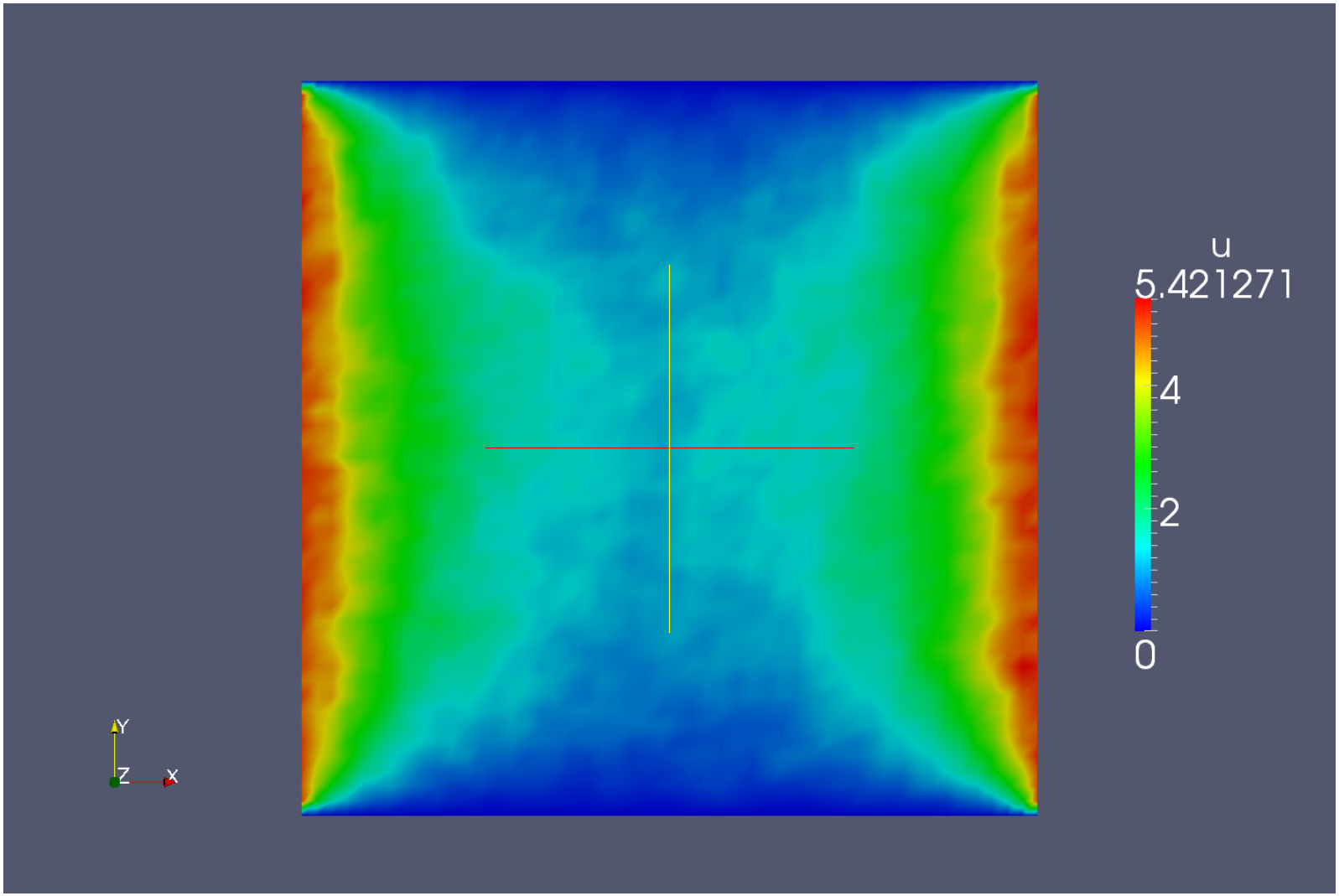


Figure : **Adjoint solution.** 1×10^5 *total histories.*

OAK RIDGE National Laboratory

Solving the Heat Equation: Adjoint Method

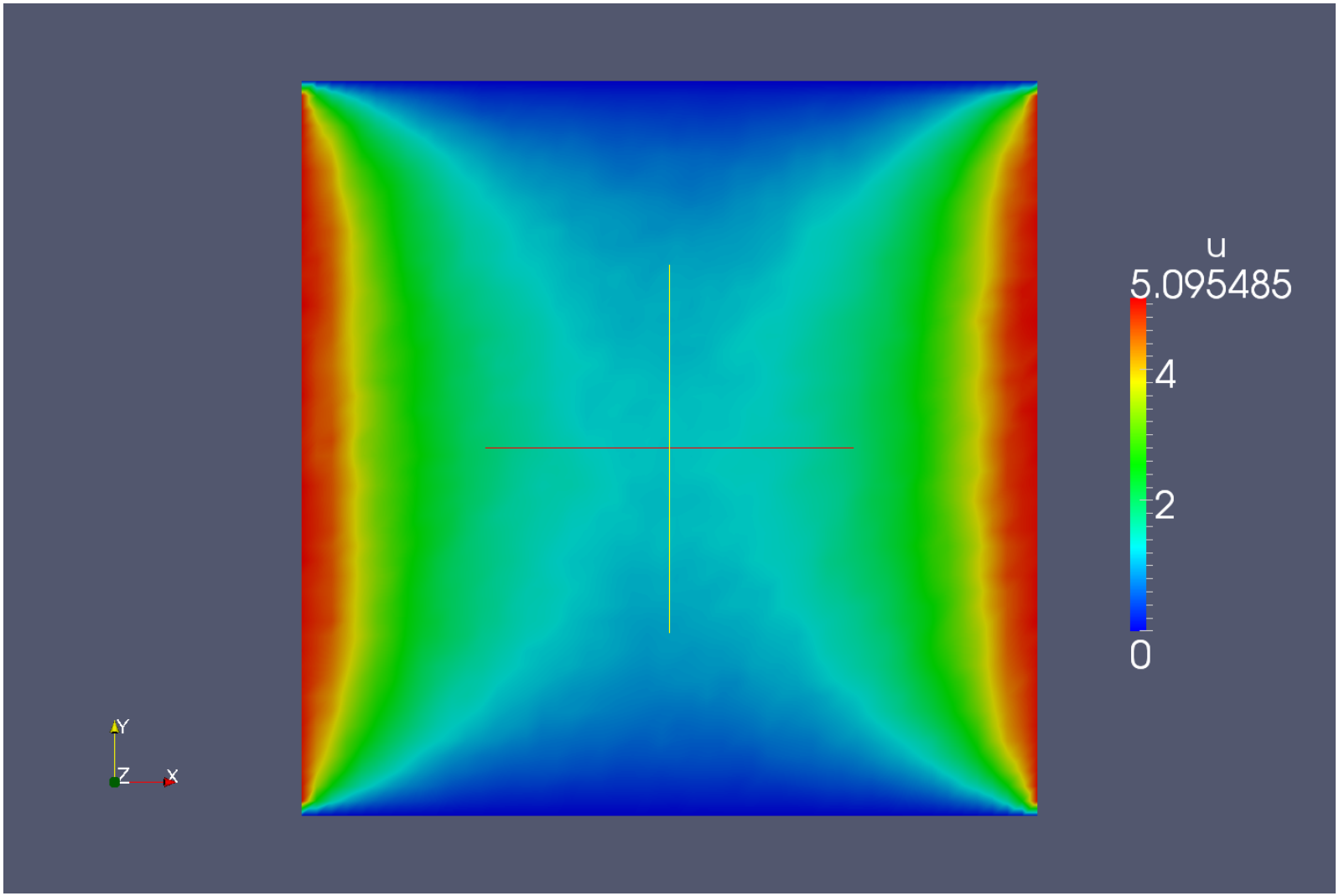


Figure : **Adjoint solution.** 1×10^6 *total histories.*

OAK RIDGE National Laboratory

Solving the Heat Equation: Adjoint Method

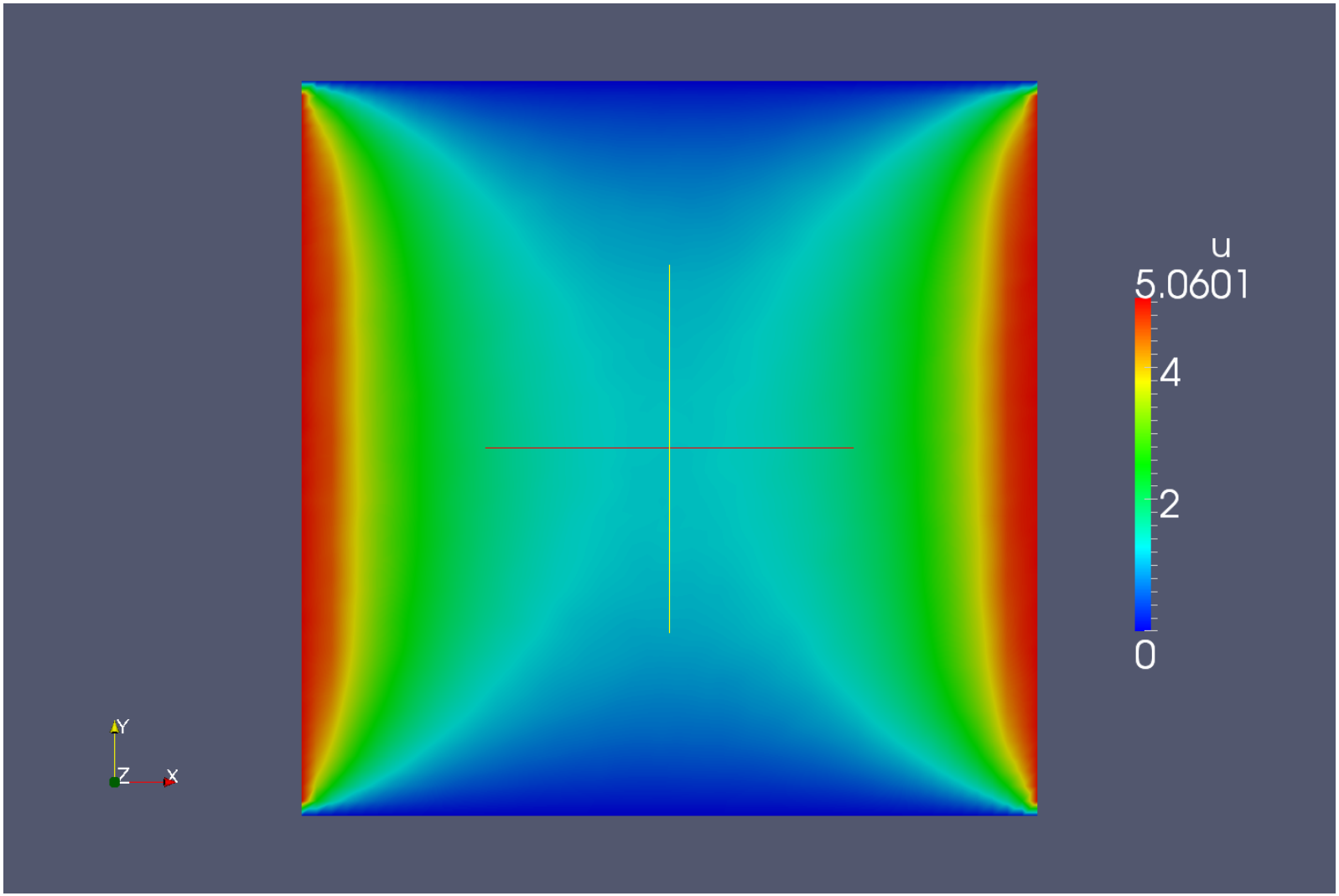
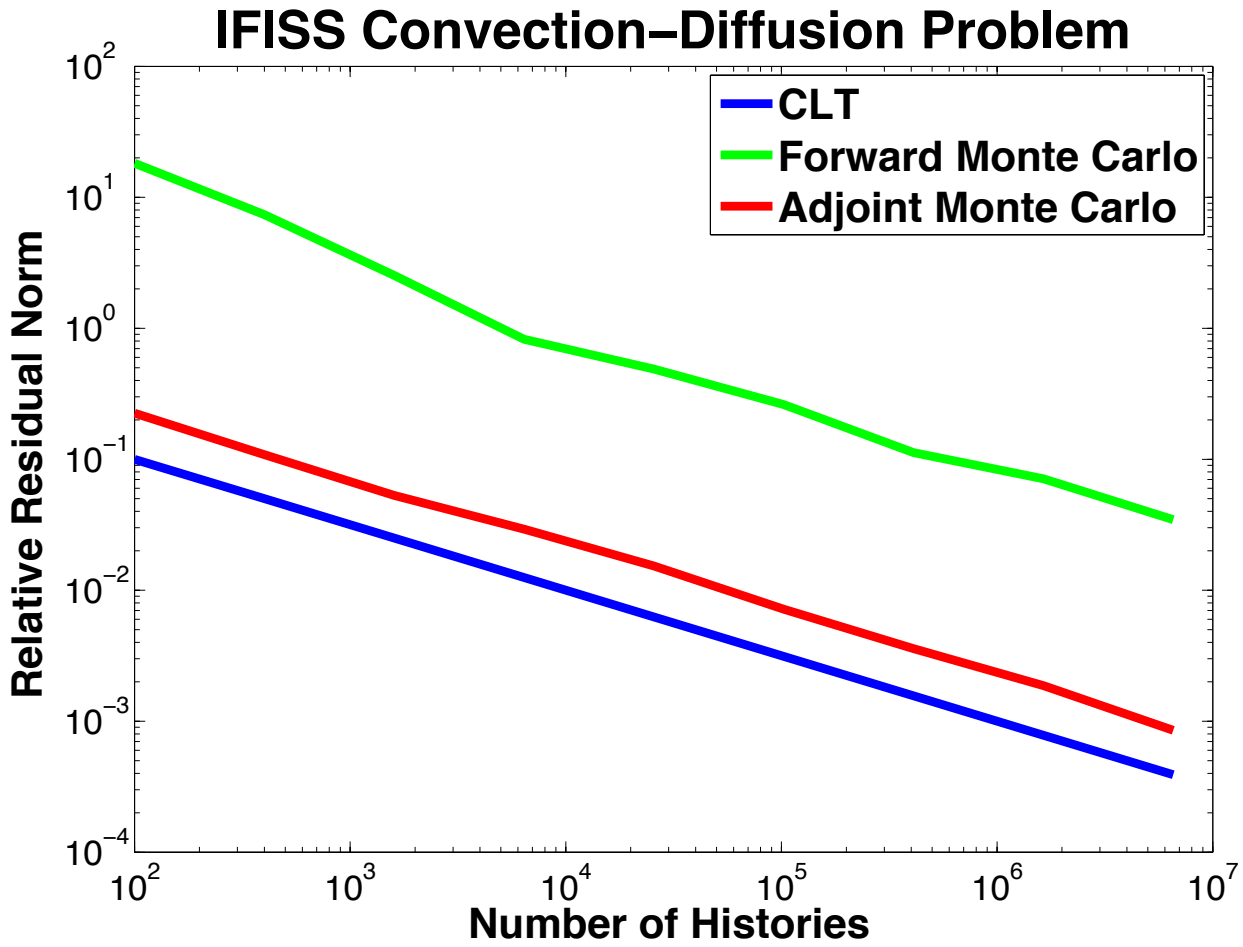


Figure : **Adjoint solution.** 1×10^7 *total histories.*

OAK RIDGE National Laboratory

Limitations of Monte Carlo

- Solving linear systems of equations with "pure" Monte Carlo methods is generally intractable
  - Central limit theorem is barrier to accurate solutions



OAK RIDGE National Laboratory

Residual Monte Carlo Methods

Sequential Monte Carlo

- Devised by John Halton in the 1960's as a residual method
- Instead of directly solving $\mathbf{Ax} = \mathbf{b}$ with Monte Carlo, apply Monte Carlo to residual equation $\mathbf{A}\delta = \mathbf{r}$
- Preconditioned Richardson iteration using Monte Carlo as preconditioner:

$$\begin{aligned} \mathbf{r}^k &= \mathbf{b} - \mathbf{A}\mathbf{x}^k \\ \hat{\mathbf{A}}\delta &= \mathbf{r}^k \\ \mathbf{x}^{k+1} &= \mathbf{x}^k + \delta \end{aligned}$$

- Exponential convergence achieved by decoupling Monte Carlo error from solution error

OAK RIDGE National Laboratory

Monte Carlo Synthetic Acceleration

- Devised by Evans and Mosher in the 2000's as an acceleration scheme for radiation diffusion problems (LANL)
- Can be abstracted as a general linear solver
- Combine with Richardson iteration as a "smoother" in between Monte Carlo steps:

$$\begin{aligned}
\mathbf{r}^k &= \mathbf{b} - \mathbf{A}\mathbf{x}^k \\
\mathbf{x}^{k+1/2} &= \mathbf{x}^k + \mathbf{r}^k \\
\mathbf{r}^{k+1/2} &= \mathbf{b} - \mathbf{A}\mathbf{x}^{k+1/2} \\
\hat{\mathbf{A}}\delta &= \mathbf{r}^{k+1/2} \\
\mathbf{x}^{k+1} &= \mathbf{x}^{k+1/2} + \delta
\end{aligned}$$

OAK RIDGE National Laboratory

Preconditioned MCSA

$$\mathbf{r}^k = \mathbf{M}_L^{-1}(\mathbf{b} - \mathbf{A}\mathbf{M}_R^{-1}\mathbf{x}^k)$$

$$\mathbf{x}^{k+1/2} = \mathbf{x}^k + \mathbf{r}^k$$

$$\mathbf{r}^{k+1/2} = \mathbf{M}_L^{-1}(\mathbf{b} - \mathbf{A}\mathbf{M}_R^{-1}\mathbf{x}^{k+1/2})$$

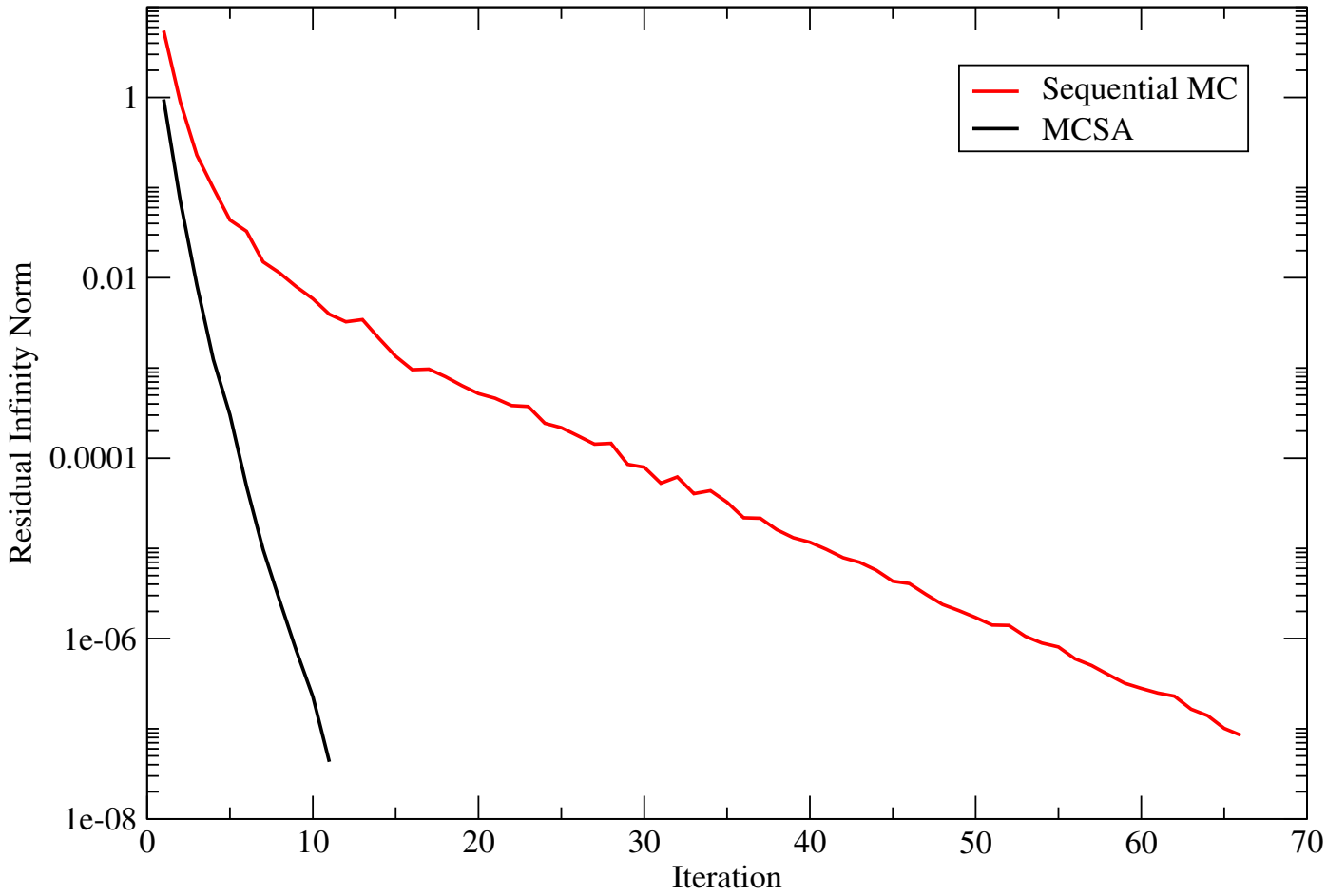
$$\mathbf{M}_L^{-1}\mathbf{A}\mathbf{M}_R^{-1}\delta\mathbf{x}^{k+1/2} = \mathbf{r}^{k+1/2}$$

$$\mathbf{x}^{k+1} = \mathbf{x}^{k+1/2} + \delta\mathbf{x}^{k+1/2}$$

- Requires explicit construction of preconditioned matrix
  - This has generally limited preconditioner selection to diagonal, block diagonal, sparse approximate inverse approaches
- Sparsity in $\mathbf{A}$, $\mathbf{M}$, does not imply sparsity in $\mathbf{M}^{-1}\mathbf{A}$
- Need to investigate preconditioning strategies that lead to sparse preconditioned systems (Michele Benzi, Emory)

OAK RIDGE National Laboratory

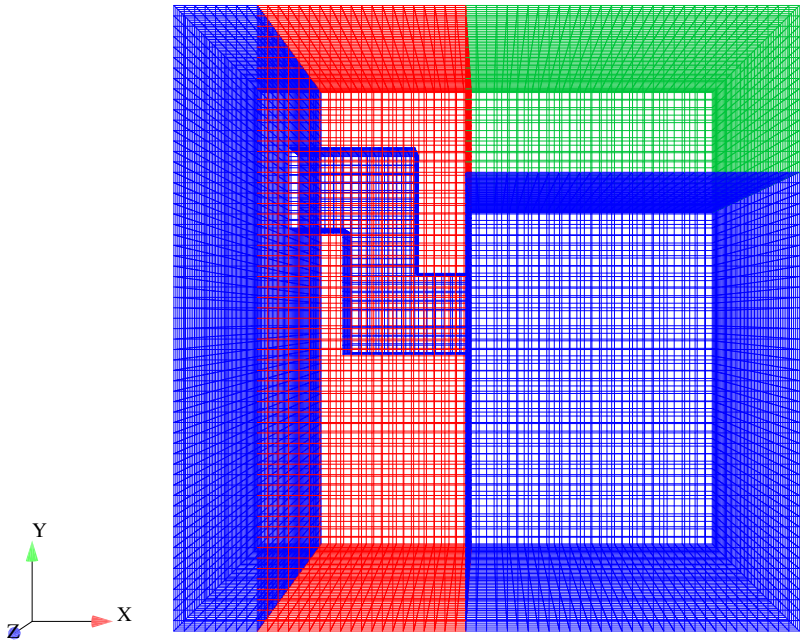
Residual Algorithm Comparison

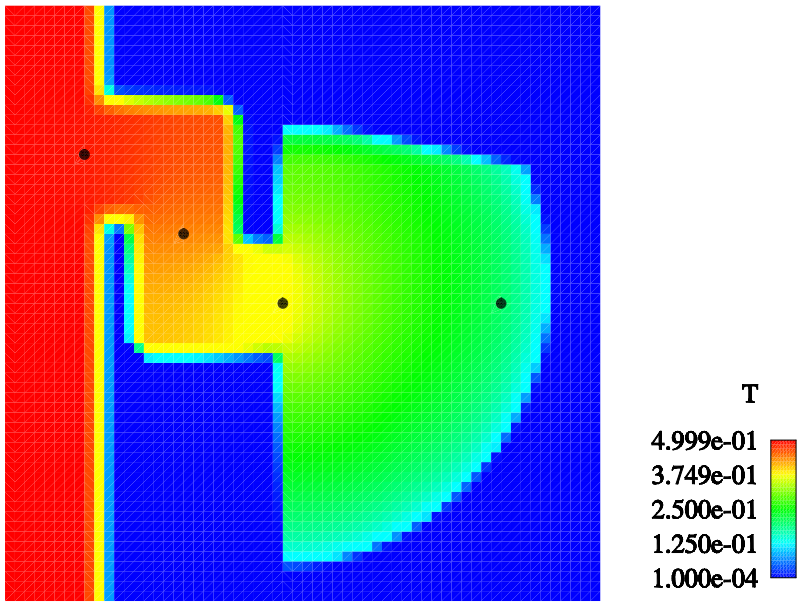


OAK RIDGE National Laboratory

3D Equilibrium Thermal Radiation Diffusion

T. Evans, S. Mosher, S. Slattery, S. Hamilton, "A Monte Carlo synthetic-acceleration method for solving the thermal radiation diffusion equation," Journal of Computational Physics **258**, pp. 338–358 (2014).





Solver	Total Iterations	Relative Time Per Iteration	Relative Total Time
CG (Jacobi)	42,881	2.93	1.02
CG (ML)	14,097	26.60	1.94
MCSA (Jacobi)	90,129	1.0	1.0

OAK RIDGE National Laboratory

Path to Harder Problems

- Experience with radiation diffusion suggests that for problems where spectral radius is low ($\lesssim 0.8$), MCSA can be competitive with leading methods
  - Very low cost per iteration to approximate Neumann series

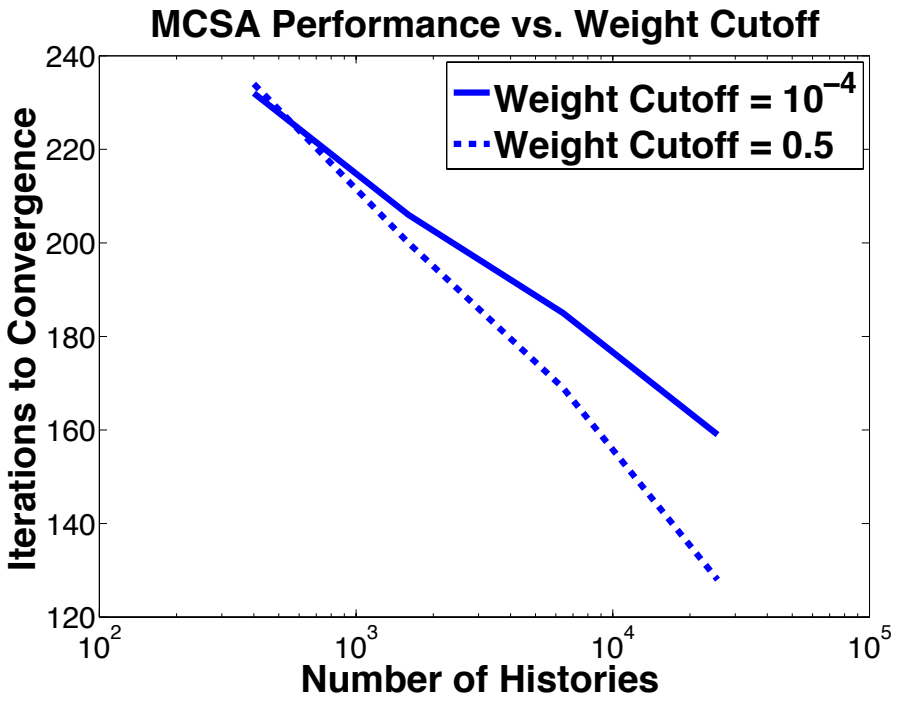
- What about more challenging problems?
  - Neumann series converges more slowly, so histories last longer
  - More histories may be required
  - What if Neumann series is not convergent?

OAK RIDGE National Laboratory

Polynomial Formulation

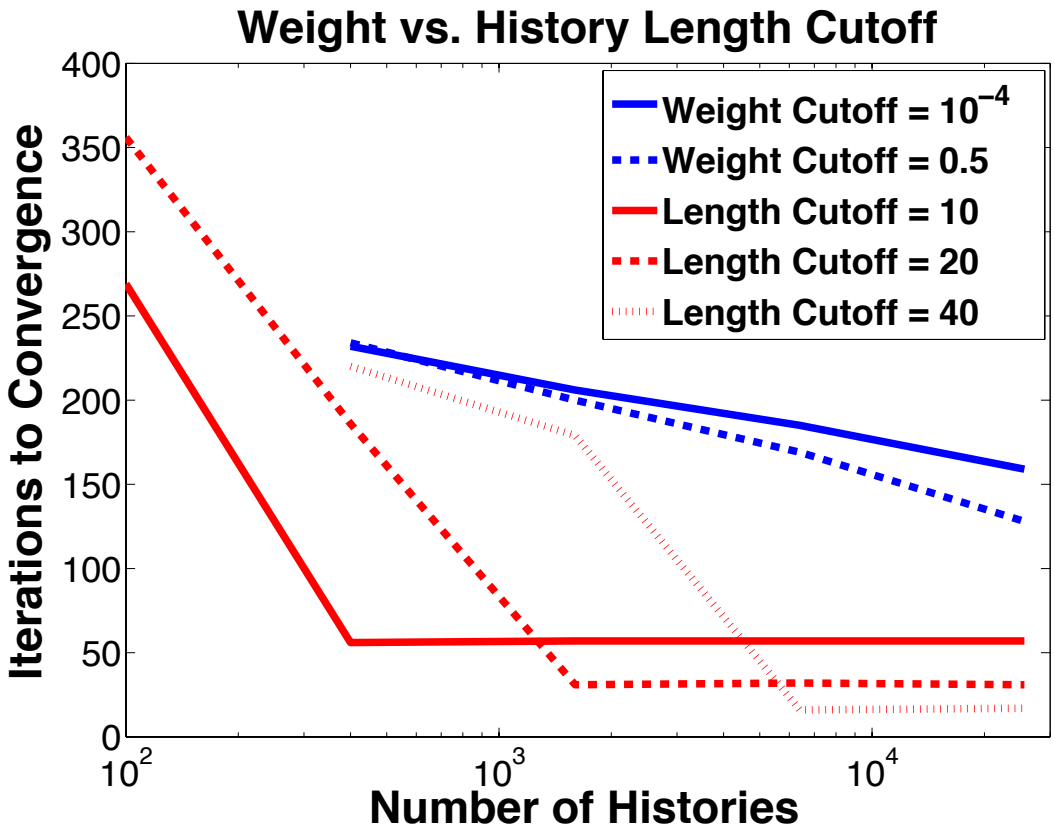
Stopping Criteria – Weight Cutoff

- Histories are usually terminated when the relative weight drops below a specified cutoff (i.e. $\frac{w}{w_0} < \tau$)
- Adjoint MCSA on JPWH_991 matrix with diagonal preconditioning ($\rho(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}) \approx 0.98$):



OAK RIDGE National Laboratory

Weight vs. History Length Cutoff



- Unlike weight cutoff, history length cutoff "saturates" at some number of histories – fixed length Neumann series has been effectively reproduced
- Saturation point is higher for longer history lengths

OAK RIDGE National Laboratory

Monte Carlo as Approximate Polynomial Preconditioning

- Using history length cutoff, Monte Carlo process is approximating a fixed length Neumann series polynomial

$$x_i = \sum_{k=0}^{\infty} \sum_{i_1}^{N} \sum_{i_2}^{N} \cdots \sum_{i_k}^{N} h_{i,i_1} h_{i_1,i_2} \ldots h_{i_{k-1},i_k} b_{i_k}$$

- As number of histories grows, iteration count becomes identical to using "true" polynomial

- Why limit ourselves to the Neumann series polynomial?
  - Chebyshev or GMRES polynomials are viable alternatives

OAK RIDGE National Laboratory

Neumann Series Polynomial

Table : Adjoint MCSA with Neumann Polynomial, 1000×1000 Shifted Laplacian Matrix. Values are MCSA iteration counts (timing in milliseconds)

Histories per Iteration	Polynomial Order		
	2	4	6
250	775(100)	651(96)	664(110)
500	725(122)	502(104)	394(95)
1000	707(174)	482(179)	366(144)
2000	703(280)	471(259)	356(251)
4000	698(494)	467(497)	350(458)
8000	697(923)	464(905)	350(873)
16000	695(1796)	462(1768)	347(1711)

OAK RIDGE National Laboratory

Chebyshev Polynomial

Table : Adjoint MCSA with Chebyshev Polynomial, 1000×1000 Shifted Laplacian Matrix. Values are MCSA iteration counts (timing in milliseconds)

Histories per Iteration	Polynomial Order		
	1	2	3
250	-	-	-
500	-	-	-
1000	-	-	-
2000	-	328(134)	-
4000	-	296(210)	-
8000	-	288(380)	262(423)
16000	1132(1866)	283(721)	175(550)

OAK RIDGE National Laboratory

Polynomial Methods – Summary

- Using history length cutoff rather than weight cutoff can lead to large reductions in iteration counts
- A good approximation to a few terms in the Neumann series performs better than a statistically noisy approximation to full series
- Significant reduction in iteration counts are possible using alternate polynomials, but generally outweighed by increase in number of histories required
  - May be very beneficial from resiliency and parallel efficiency standpoints
  - Will be re-evaluated in future efforts

OAK RIDGE National Laboratory

Parallelism

OAK RIDGE National Laboratory

Domain Decomposed Monte Carlo

- Each parallel process owns a piece of the domain (linear system)
- Random walks must be transported between adjacent domains through parallel communication
- Domain decomposition determined by the input system

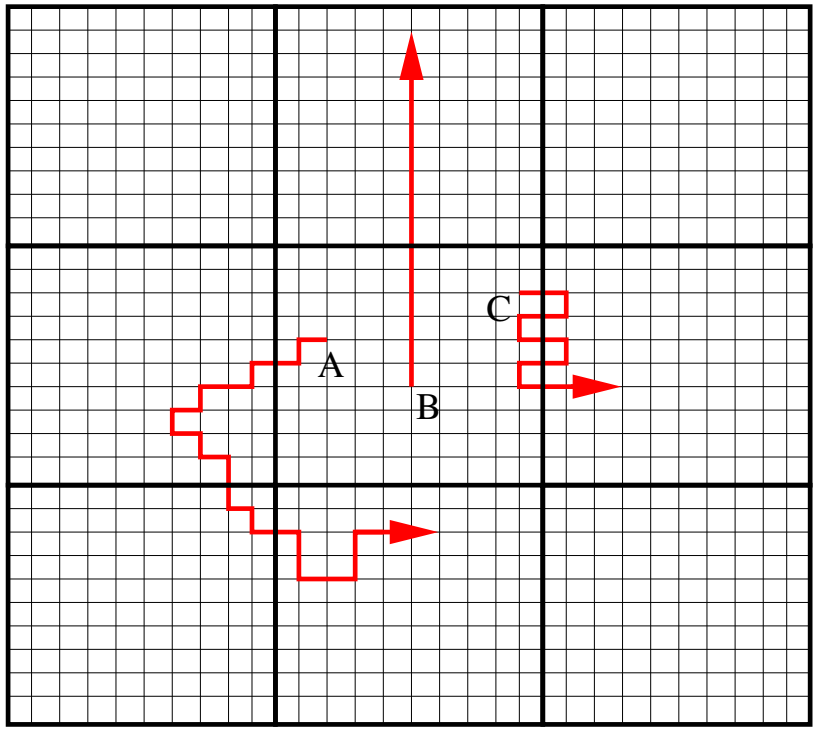
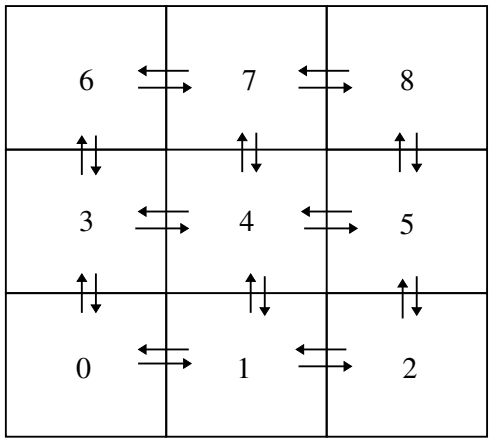


Figure : Domain decomposition example illustrating how domain-to-domain transport creates communication costs.

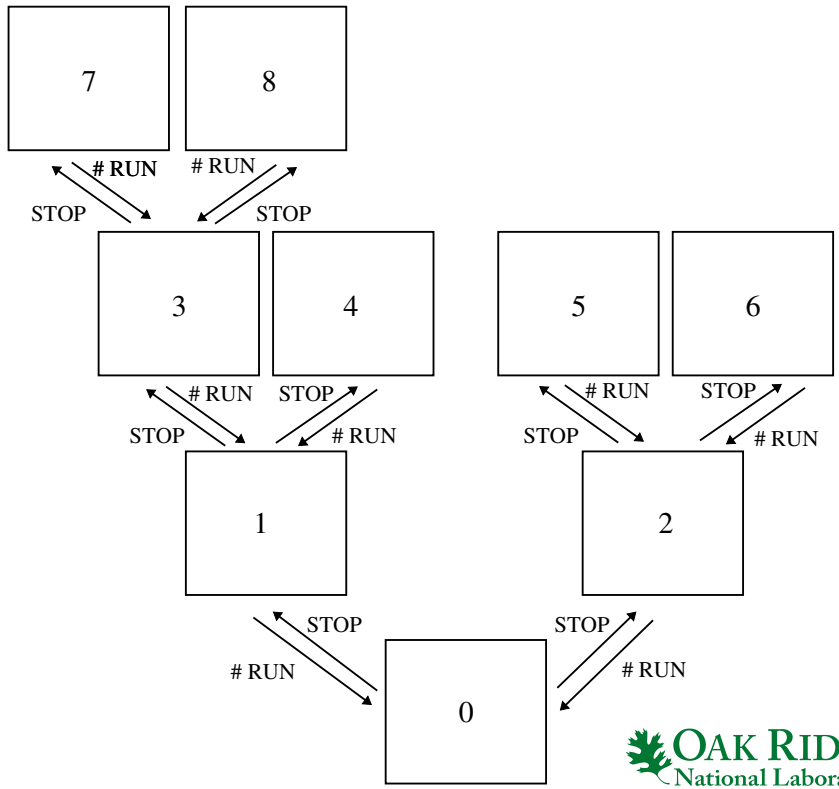
OAK RIDGE National Laboratory

Asynchronous Monte Carlo Transport Kernel

- General extension of the Milagro algorithm (LANL)
- Asynchronous nearest neighbor communication of histories
- Binary asynchronous communication tree for completing transport



- Extensible to problems where histories may be created (i.e. variance reduction)



OAK RIDGE National Laboratory

Parallel Application – Nuclear Reactor Analysis

The simplified P_N (SP_N) equations are an approximation to the Boltzmann neutron transport equation used to simulate nuclear reactors

$$\hat{\Omega} \cdot \vec{\nabla}\psi(\vec{r}, \hat{\Omega}, E) + \sigma(\vec{r}, E)\psi(\vec{r}, \hat{\Omega}, E) = \iint \sigma_s(\vec{r}, E' \to E, \hat{\Omega}' \cdot \hat{\Omega})\psi(\vec{r}, \hat{\Omega}', E')d\Omega' dE' + q(\vec{r}, \hat{\Omega}, E) \quad (1)$$

$$-\nabla \cdot \left[\frac{n}{2n+1} \frac{1}{\Sigma_{n-1}} \nabla \left(\frac{n-1}{2n-1}\phi_{n-2} + \frac{n}{2n-1}\phi_n \right) + \frac{n+1}{2n+1} \frac{1}{\Sigma_{n+1}} \nabla \left(\frac{n+1}{2n+3}\phi_n + \frac{n+2}{2n+3}\phi_{n+2} \right) \right] + \Sigma_n \phi_n = q\delta_{n0} \qquad n = 0, 2, 4, \cdots, N \quad (2)$$

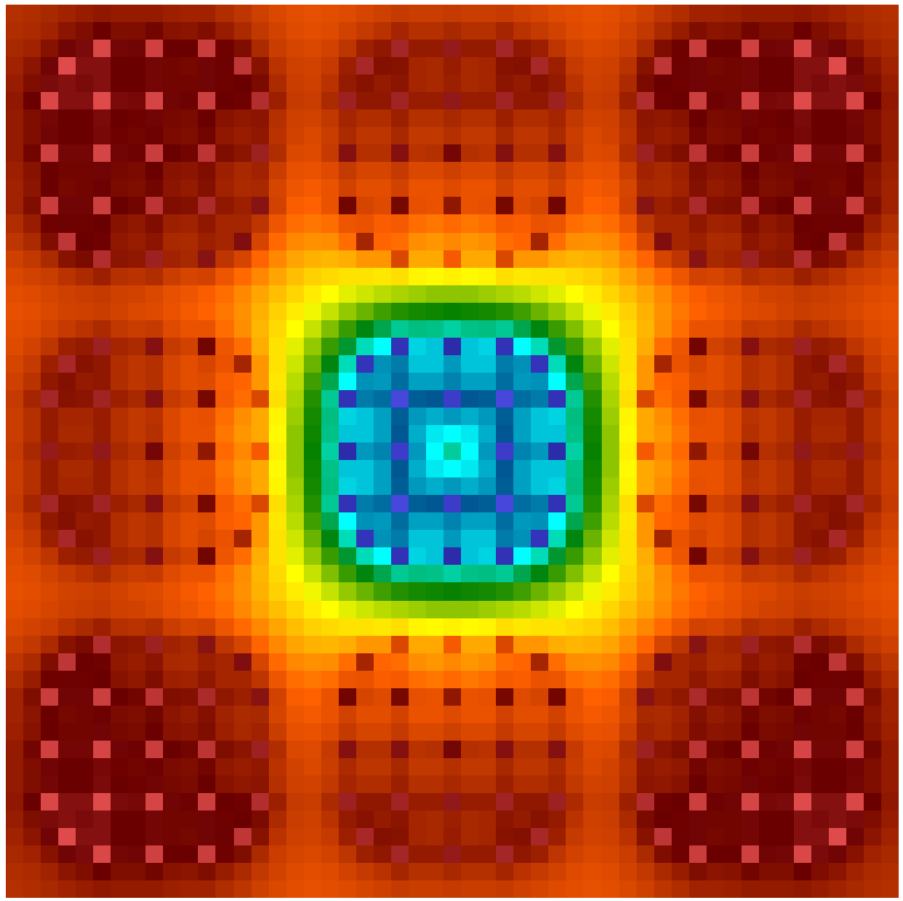
$$-\nabla \cdot \mathbb{D}_n \nabla \mathbb{U}_n + \sum_{m=1}^{4} \mathbb{A}_{nm}\mathbb{U}_m = \frac{1}{k} \sum_{m=1}^{4} \mathbb{F}_{nm}\mathbb{U}_m \qquad n = 1, 2, 3, 4 \quad (3)$$

OAK RIDGE National Laboratory

SP_N Assembly Problem

Test problem – 3×3 array of fuel assemblies with control rod in center location (Profugus)

- 23 energy groups, 2 angular moments, 25M degrees of freedom
- 1,000 computational cores via domain decomposition
- We are interested in solving generalized eigenvalue problem, for this study we use Arnoldi as the eigensolver and compare different methods for solving linear systems



OAK RIDGE National Laboratory

Parallel SP_N Results

Method	Total Linear Iteration	Setup Time (s)	Solve Time (s)
GMRES-ILUT	1675	0.7	18.4
GMRES-AMG	626	0.7	46.0
GMRES-MGE	498	1.5	33.7
Richardson-AINV	5208	20.6	52.0
MCSA-AINV	1268	25.5	46.6

- ILUT preconditioning is winner here, but known to have issues with parallel scaling on large core counts
- Solve times for MCSA are competitive, but setup times are very large due to construction of sparse approximate inverse

OAK RIDGE National Laboratory

Alternative Parallelism – Additive Schwarz

- Instead of performing Monte Carlo on full problem, another possibility is to apply Monte Carlo as an additive Schwarz approach
- Decompose problem into (possibly overlapping) domains
- Perform Monte Carlo on individual subdomains
  - No communication costs in Monte Carlo problem!
- With domain decomposed Monte Carlo, iteration counts are effectively independent of the number of processors
- In an additive Schwarz approach, the preconditioner will become less effective as processor counts grow – algorithmic scalability may be an issue
- **Replication for resiliency and performance**

OAK RIDGE National Laboratory

More Parallelism – Threading

- Within a Monte Carlo solve, every history is independent of other histories – great potential for highly concurrent hardware (GPU, Xeon Phi)
- Polynomial formulation enables a priori determination of operation counts per thread
- Memory locality an issue due to random access via random walks (block formulation?)
- Early experiments using the Trilinos Kokkos library show promising performance for multi-core CPUs
- Team members recently took part in OLCF "Hackathon" in late October to begin implementing computation kernels in OpenACC to allow for GPU capability on Titan with early results indicating **1.3-9.4x** speedup for MCSA (largely dependent on random number generation)

OAK RIDGE National Laboratory

Conclusions

Additional Thoughts

- Our current implementations rely on performing fixed number of histories per Monte Carlo solve
  - How can we dynamically select/adapt the number of histories that should be performed? Use variance-based stopping criteria?
- The "almost optimal" selection of the probability and weight matrices is arbitrary (as long as $\mathbf{H}/\mathbf{H}^T = \mathbf{P} \circ \mathbf{W}$)
  - If $\mathbf{P}$ is taken to have a uniform distribution within each row, then sampling from the distribution can be done in constant (rather than logarithmic) time
- Exploring Anderson acceleration of MCSA (Tim Kelley)

OAK RIDGE National Laboratory

Monte Carlo Linear Solvers Library (MCLS)

- Designed to be easily incorporated with production physics codes
- General asynchronous MSOD MCSA implementation
  - Forward and adjoint Monte Carlo with method of expected values
  - Parallel row matrix/vector interface
  - General fixed point iteration strategy
  - Explicit algebraic preconditioner suite
  - Planned extensions to threaded/hybrid parallelism
- Implemented in C++
- Heavy use of the Trilinos scientific computing libraries
- Open-source BSD 3-clause license
- `https://github.com/sslattery/MCLS`

OAK RIDGE National Laboratory

Profugus Neutronics Mini-Application

A mini-application provides open-source kernels that effectively capture the algorithmic features of full applications

- Extracts core computational kernels from the Exnihilo transport code suite
  - Denovo: deterministic transport solvers
  - Shift: Monte Carlo transport
  - Insilico: reactor neutronics (cross sections, geometry, input/output)
- Exnihilo is export-controlled, Profugus is not

Profugus includes:

- Fixed source and eigenvalue problems
- Trilinos solvers/preconditioners
- Generic material and cross section libraries
- `https://github.com/ORNL-CEES/Profugus`

OAK RIDGE National Laboratory

Conclusions

- Monte Carlo methods offer great potential for both resilient and highly parallel solvers
- For certain classes of problems, Monte Carlo methods can be competitive with leading modern solvers
- Extending methods to broader problem areas is significant challenge and an attractive area for continued research
- Performance modeling and resiliency simulations this FY

OAK RIDGE National Laboratory