# Multilevel Monte Carlo Solvers for Linear Systems

Stuart Slattery, Thomas Evans, Steven Hamilton

slatterysr@ornl.gov

Oak Ridge National Laboratory

April 9, 2014

# Hardware-Based Motivation

- Modern hardware is moving in two directions (Kogge,2011):
  - Lightweight machines
  - Heterogeneous machines
  - Both characterized by low power and high concurrency
- Some issues:
  - Higher potential for both soft and hard failures (DOE,2012)
  - Memory restrictions are expected with a continued decrease in memory/FLOPS
- Potential resolution from Monte Carlo:
  - Soft failures buried within the tally variance
  - Hard failures mitigated by replication
  - Memory savings over conventional methods

# Monte Carlo Methods for Discrete Linear Systems

- First proposed by J. Von Neumann and S.M. Ulam in the 1940's
- Earliest published reference in 1950 (Forsythe,1950)
- General lack of publications on applications
- Plagued by slow convergence $\approx \frac{1}{\sqrt{N}}$
- Modern work yielded new applications (Evans,2009) (Evans,2013)

# Monte Carlo Synthetic-Acceleration

## MCSA Iteration

$$\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k$$

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

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

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

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

- Neumann-Ulam methods bound by the Central Limit Theorem
- Build on Sequential Monte Carlo method (Halton,1962)
- Neumann-Ulam Monte Carlo solver computes the correction
- Decouples MC error from solution error, exponential convergence

# Monte Carlo Linear Solver Preliminaries

- Split the linear operator

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \rightarrow \quad \mathbf{x} = \mathbf{H}\mathbf{x} + \mathbf{b}$$

$$\mathbf{H} = \mathbf{I} - \mathbf{A}$$

- Generate the *Neumann series*

$$\mathbf{A}^{-1} = (\mathbf{I} - \mathbf{H})^{-1} = \sum_{k=0}^{\infty} \mathbf{H}^k$$

- Require $\rho(\mathbf{H}) < 1$ for convergence

$$\mathbf{A}^{-1}\mathbf{b} = \sum_{k=0}^{\infty} \mathbf{H}^k \mathbf{b} = \mathbf{x}$$

# Monte Carlo Linear Solver Preliminaries

- Expand the Neumann series

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

- Define a sequence of state transitions

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

- Define the *Neumann-Ulam decomposition*[1]

$$\mathbf{H} = \mathbf{P} \circ \mathbf{W}$$

[1] The Hadamard product $\mathbf{A} = \mathbf{B} \circ \mathbf{C}$ is defined element-wise as $a_{ij} = b_{ij} c_{ij}$.

# Direct Method

- Compute row-normalized transition probabilities and weights

$$p_{ij} = \frac{|h_{ij}|}{\sum_j |h_{ij}|}, \quad w_{ij} = \frac{h_{ij}}{p_{ij}}$$

- Generate an expectation value for the solution

$$W_m = w_{i,i_1} w_{i_1,i_2} \cdots w_{i_{m-1},i_m}$$

$$X_\nu(i_0 = i) = \sum_{m=0}^{k} W_m b_{i_m}$$

# Direct Method

- Compute the probability of a particular random walk permutation

$$P_\nu = p_{i,i_1} p_{i_1,i_2} \cdots p_{i_{k-1},i_k}$$

- Generate the estimator

$$E\{X(i_0 = i)\} = \sum_\nu P_\nu X_\nu$$

- Check that we recover the exact solution

$$\begin{aligned} E\{X(i_0 = i)\} &= \sum_{k=0}^{\infty} \sum_{i_1}^{N} \sum_{i_2}^{N} \cdots \sum_{i_k}^{N} p_{i,i_1} p_{i_1,i_2} \cdots p_{i_{k-1},i_k} w_{i,i_1} w_{i_1,i_2} \cdots w_{i_{k-1},i_k} b_{i_k} \\ &= x_i \end{aligned}$$

# Evolution of a Solution

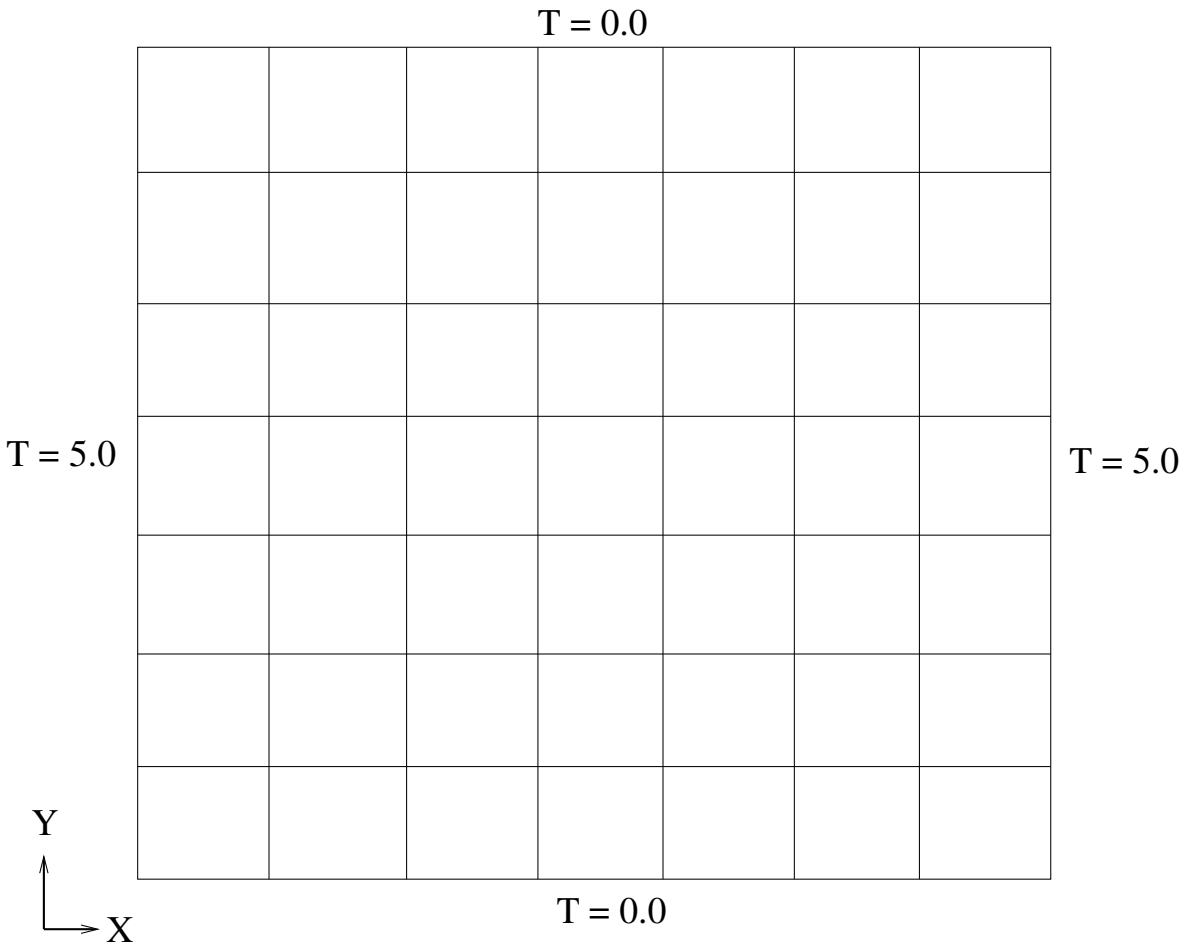


Figure : **Poisson Problem.** *Distributed source of 1.0 in the domain.*

# Evolution of a Solution: Direct Method

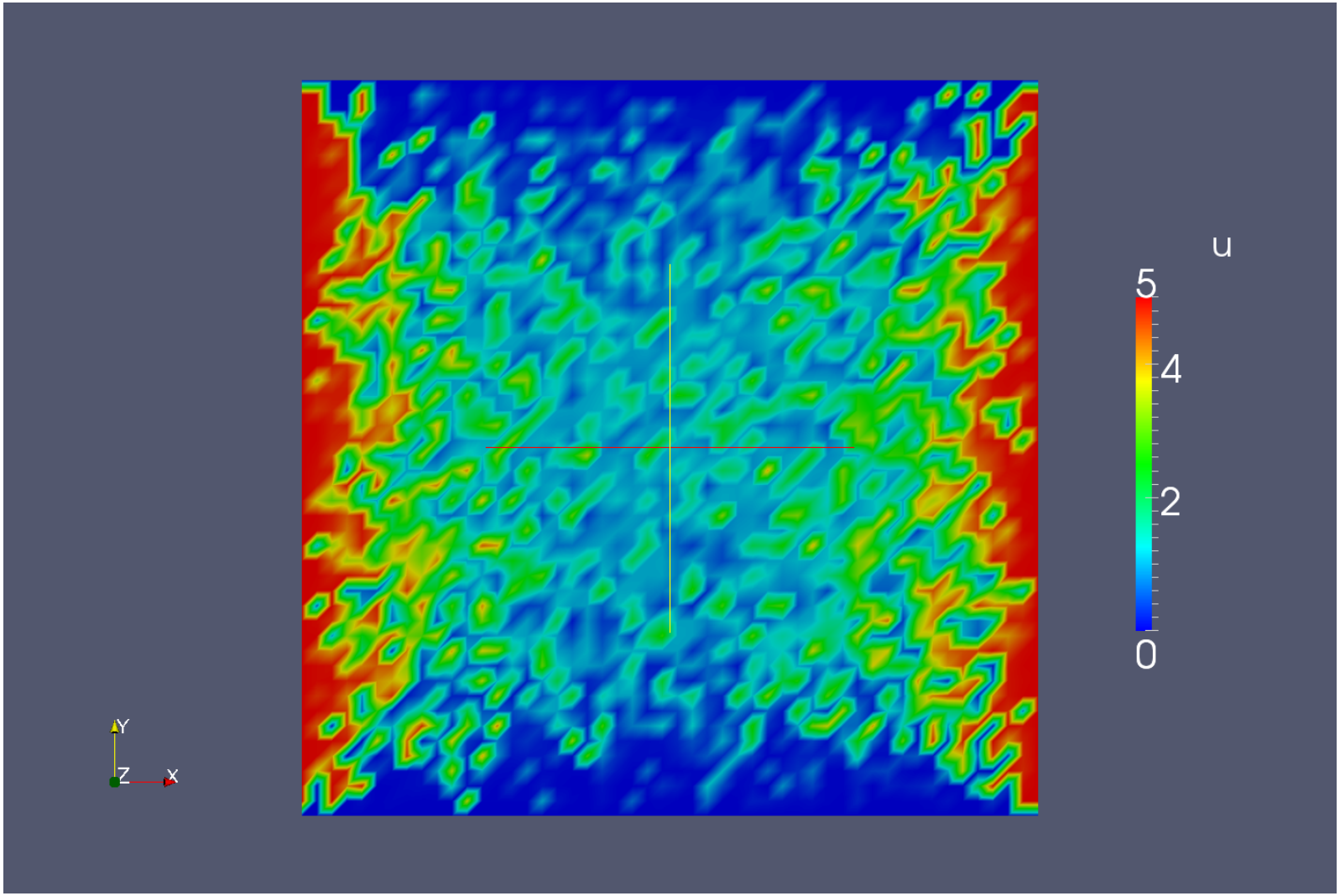


Figure : **Direct solution to Poisson Equation.** $1 \times 10^0$ *total histories.*

# Evolution of a Solution: Direct Method

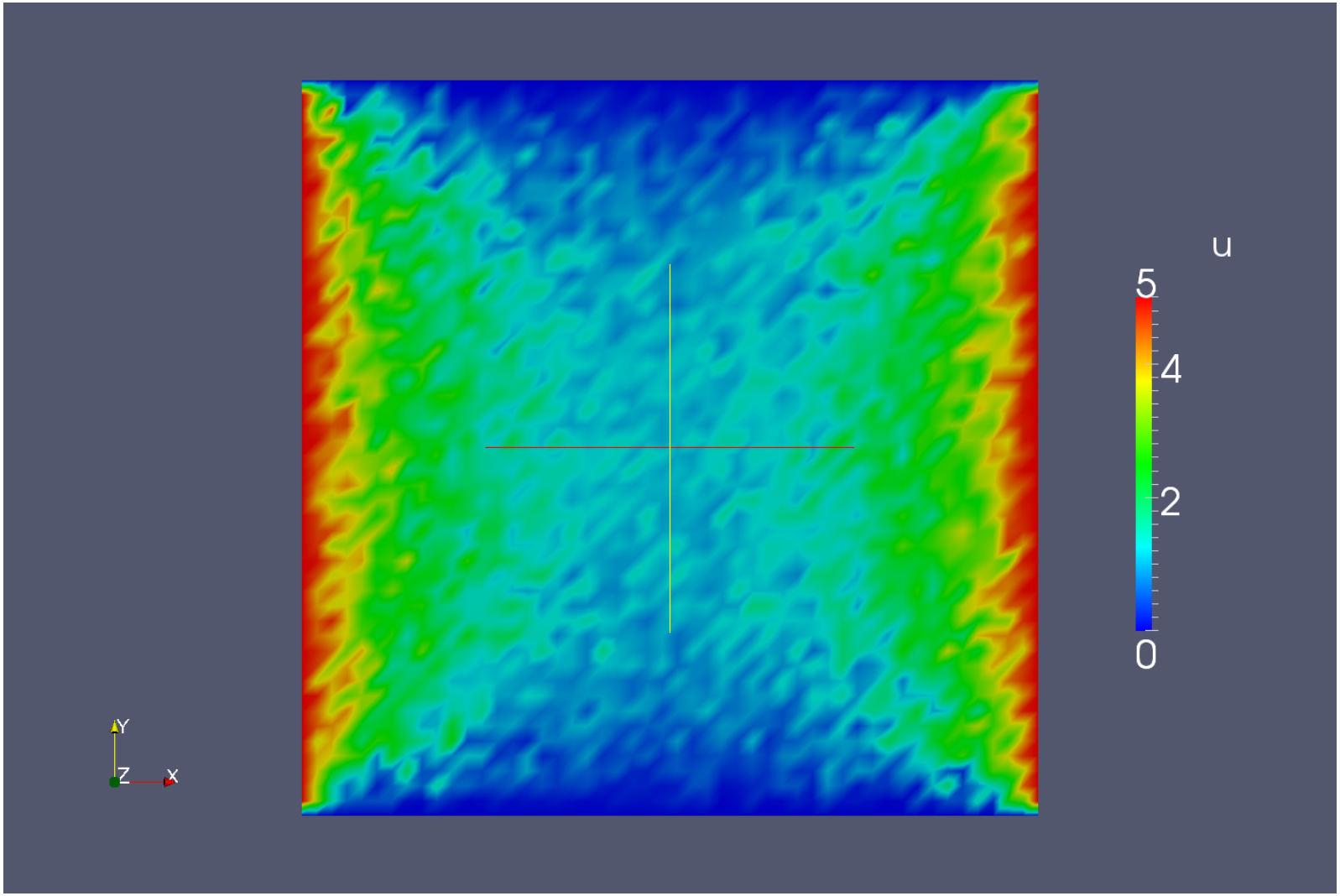


Figure : **Direct solution to Poisson Equation.** $1 \times 10^1$ *total histories.*

# Evolution of a Solution: Direct Method

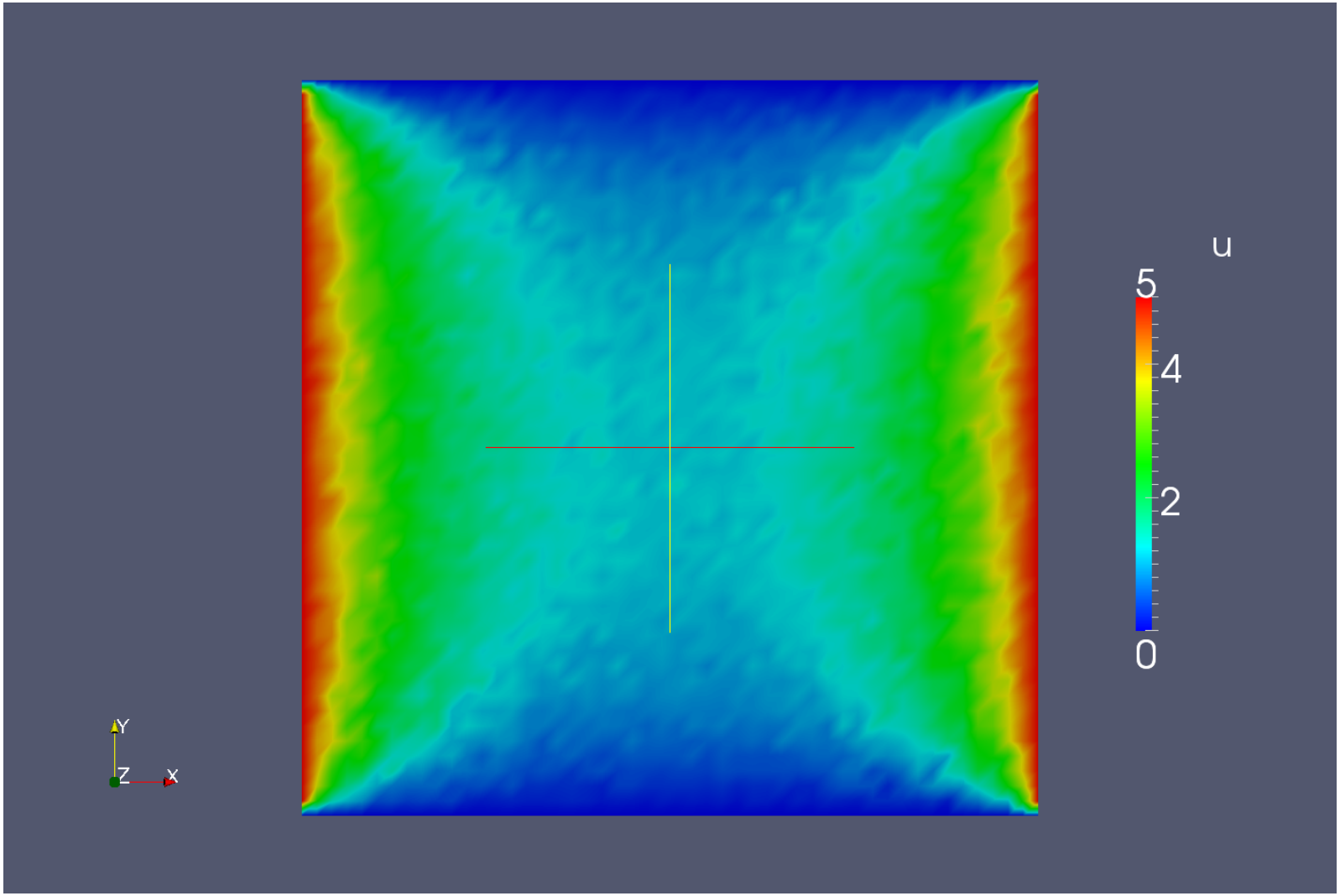


Figure : **Direct solution to Poisson Equation.** $1 \times 10^2$ *total histories.*

# Evolution of a Solution: Direct Method

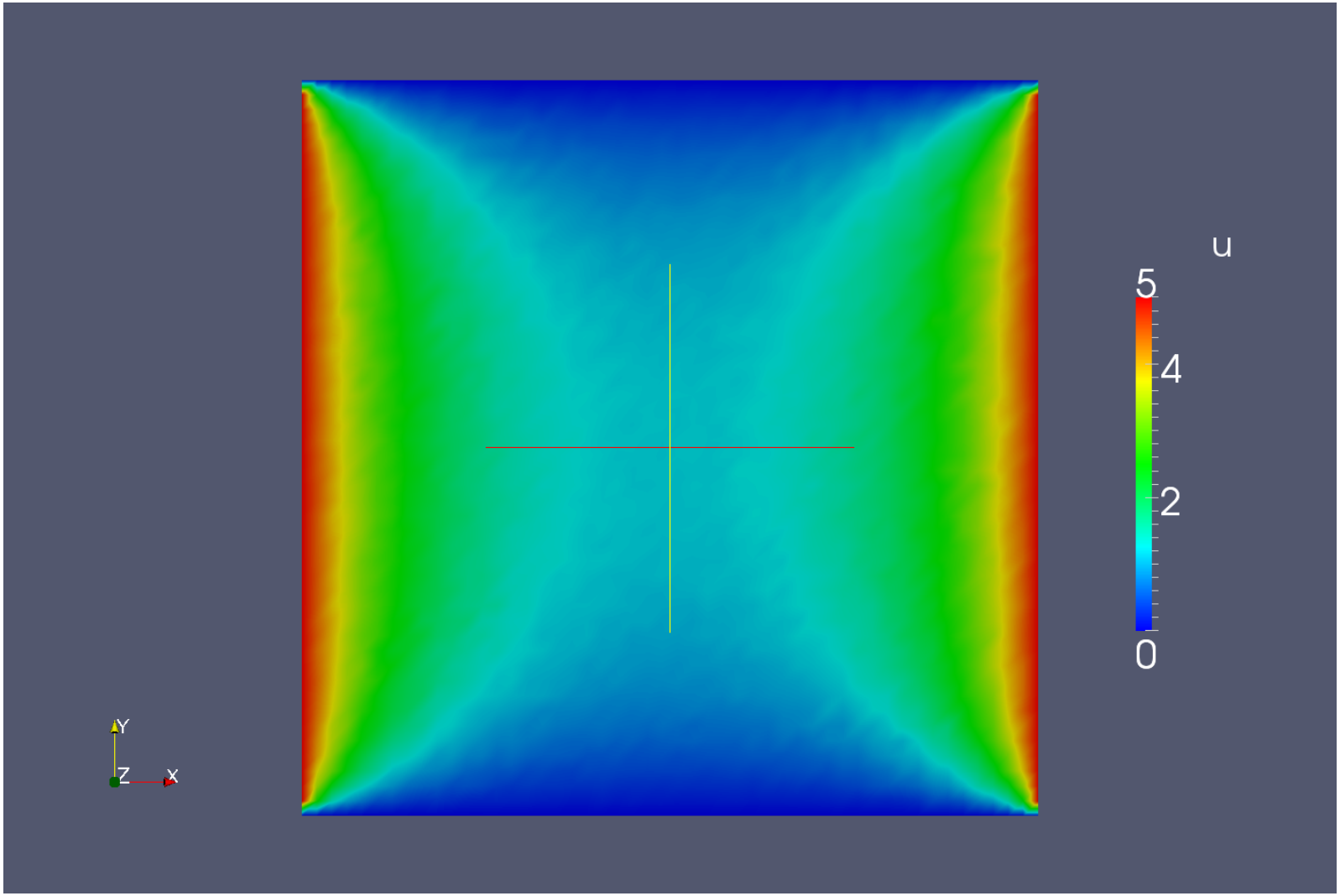


Figure : **Direct solution to Poisson Equation.** $1 \times 10^3$ *total histories.*

# Adjoint Method

- Solve the adjoint linear system

$$\mathbf{A}^T\mathbf{y} = \mathbf{d}$$

$$\mathbf{y} = \mathbf{H}^T\mathbf{y} + \mathbf{d}$$

- Set the adjoint constraint

$$\langle \mathbf{A}^T\mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{A}\mathbf{y} \rangle$$

$$\langle \mathbf{x}, \mathbf{d} \rangle = \langle \mathbf{y}, \mathbf{b} \rangle$$

# Adjoint Method

- Generate the Neumann series for the adjoint operator

$$\mathbf{y} = (\mathbf{I} - \mathbf{H}^T)^{-1}\mathbf{d} = \sum_{k=0}^{\infty}(\mathbf{H}^T)^k\mathbf{d}$$

- Expand the series

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

- Pick another constraint to yield the original solution

$$\mathbf{d} = \boldsymbol{\delta}_i,\ \langle \mathbf{y}, \mathbf{b}\rangle = \langle \mathbf{x}, \boldsymbol{\delta}_i\rangle = x_i$$

# Adjoint Method

- Use the adjoint Neumann-Ulam decomposition

$$\mathbf{H}^T = \mathbf{P} \circ \mathbf{W}$$

$$p_{ij} = \frac{|h_{ji}|}{\sum_j |h_{ji}|}, \quad w_{ij} = \frac{h_{ji}}{p_{ij}}$$

- Build the estimator and expectation value

$$X_\nu = \sum_{m=0}^{k} W_m \delta_{i,i_m}$$

$$\begin{aligned} E\{X_j\} &= \sum_{k=0}^{\infty} \sum_{i_1}^{N} \sum_{i_2}^{N} \cdots \sum_{i_k}^{N} b_{i_0} h_{i_0,i_1} h_{i_1,i_2} \ldots h_{i_{k-1},i_k} \delta_{i_k,j} \\ &= x_j \end{aligned}$$

# Evolution of a Solution: Adjoint Method

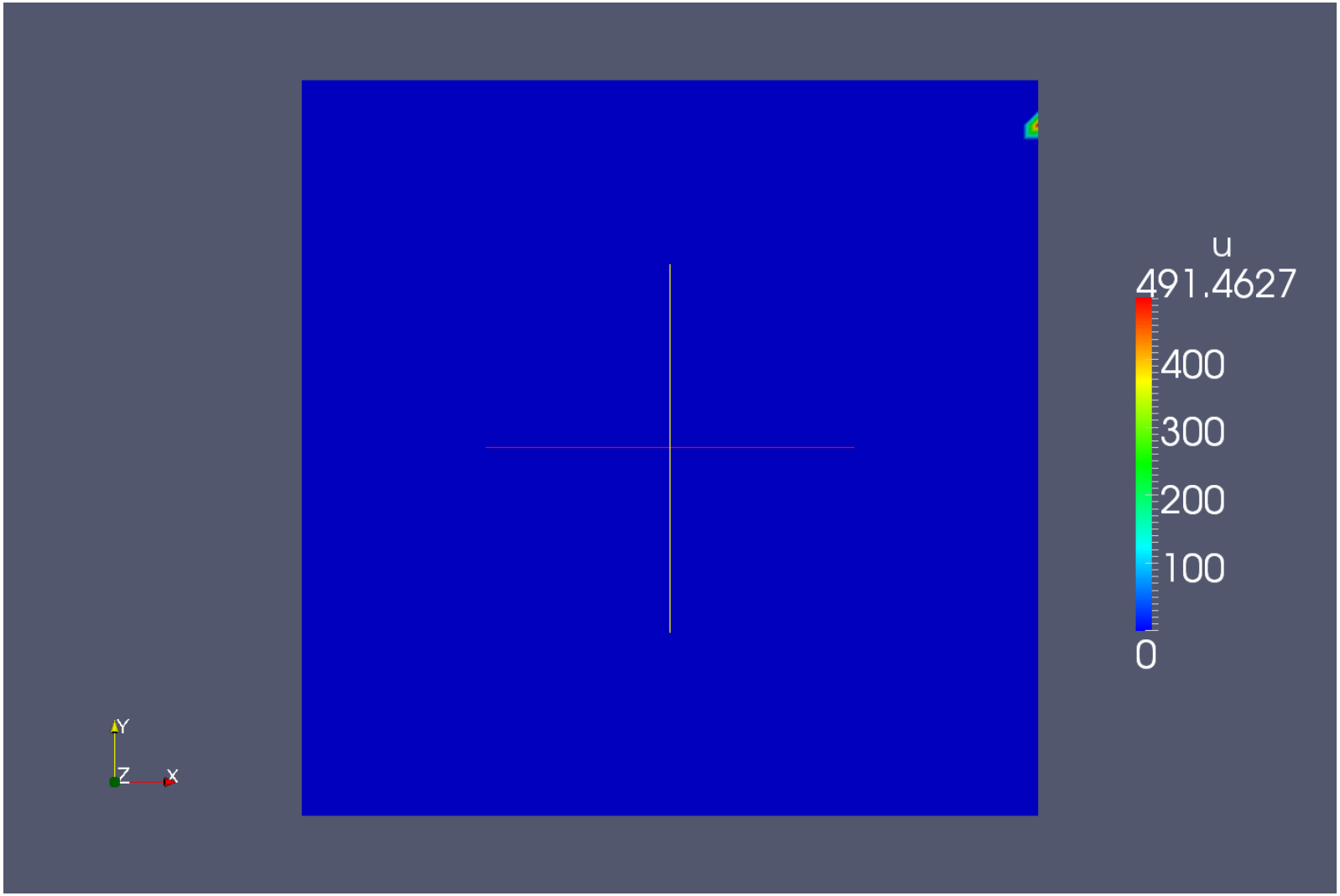


Figure : **Adjoint solution to Poisson Equation.** $1 \times 10^0$ *total histories.*

# Evolution of a Solution: Adjoint Method

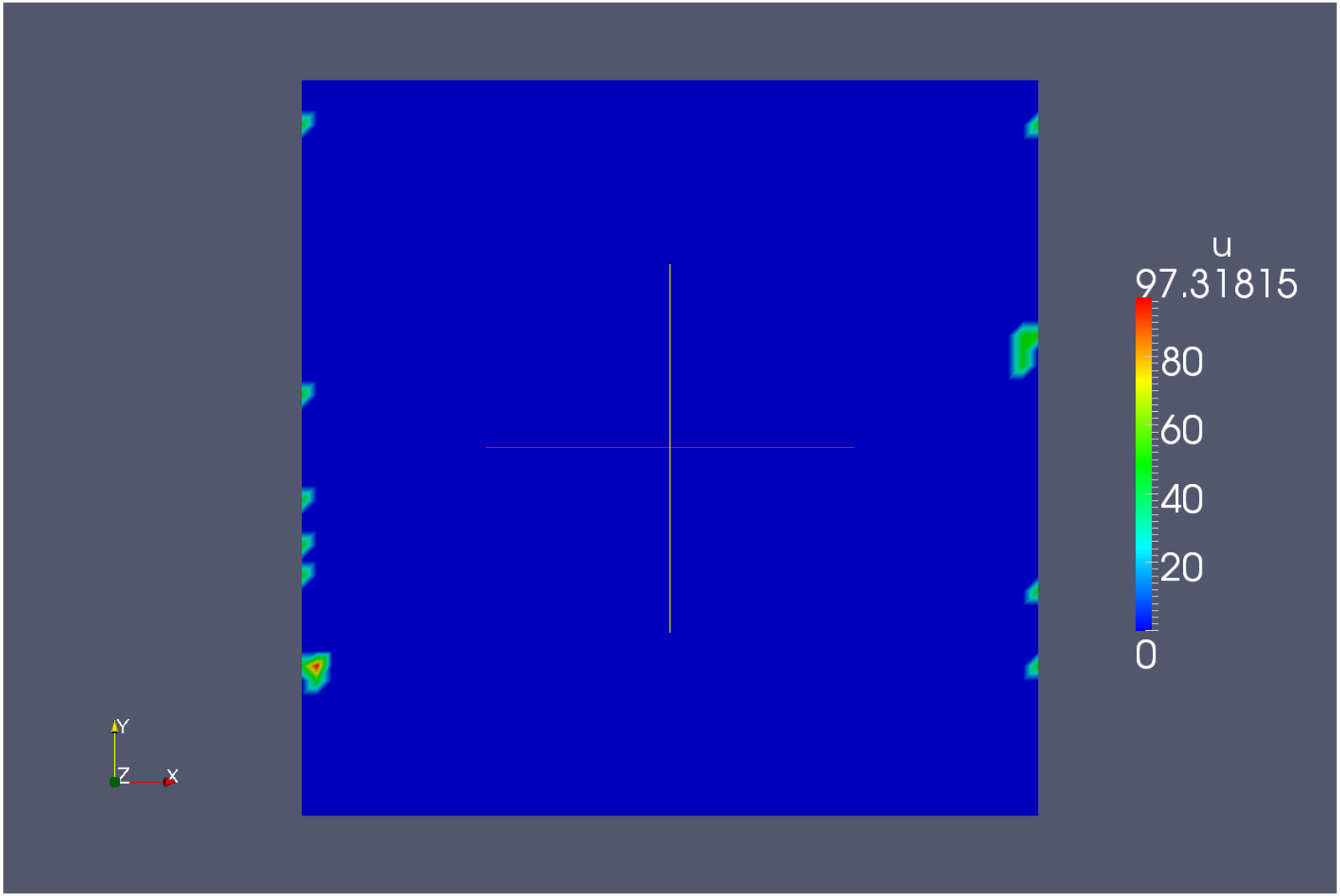


Figure : **Adjoint solution to Poisson Equation.** $1 \times 10^1$ *total histories.*

# Evolution of a Solution: Adjoint Method

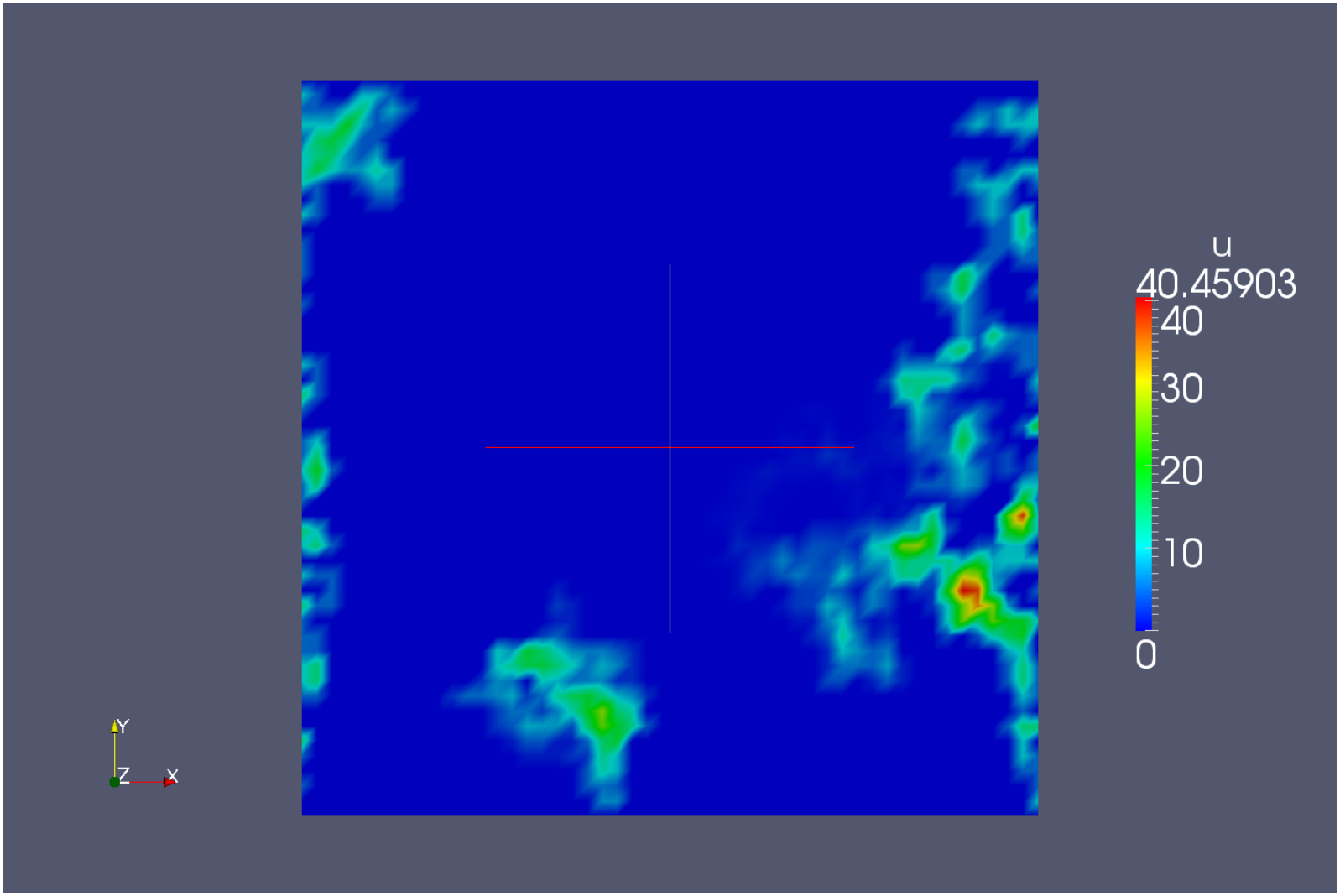


Figure : **Adjoint solution to Poisson Equation.** $1 \times 10^2$ *total histories.*

# Evolution of a Solution: Adjoint Method

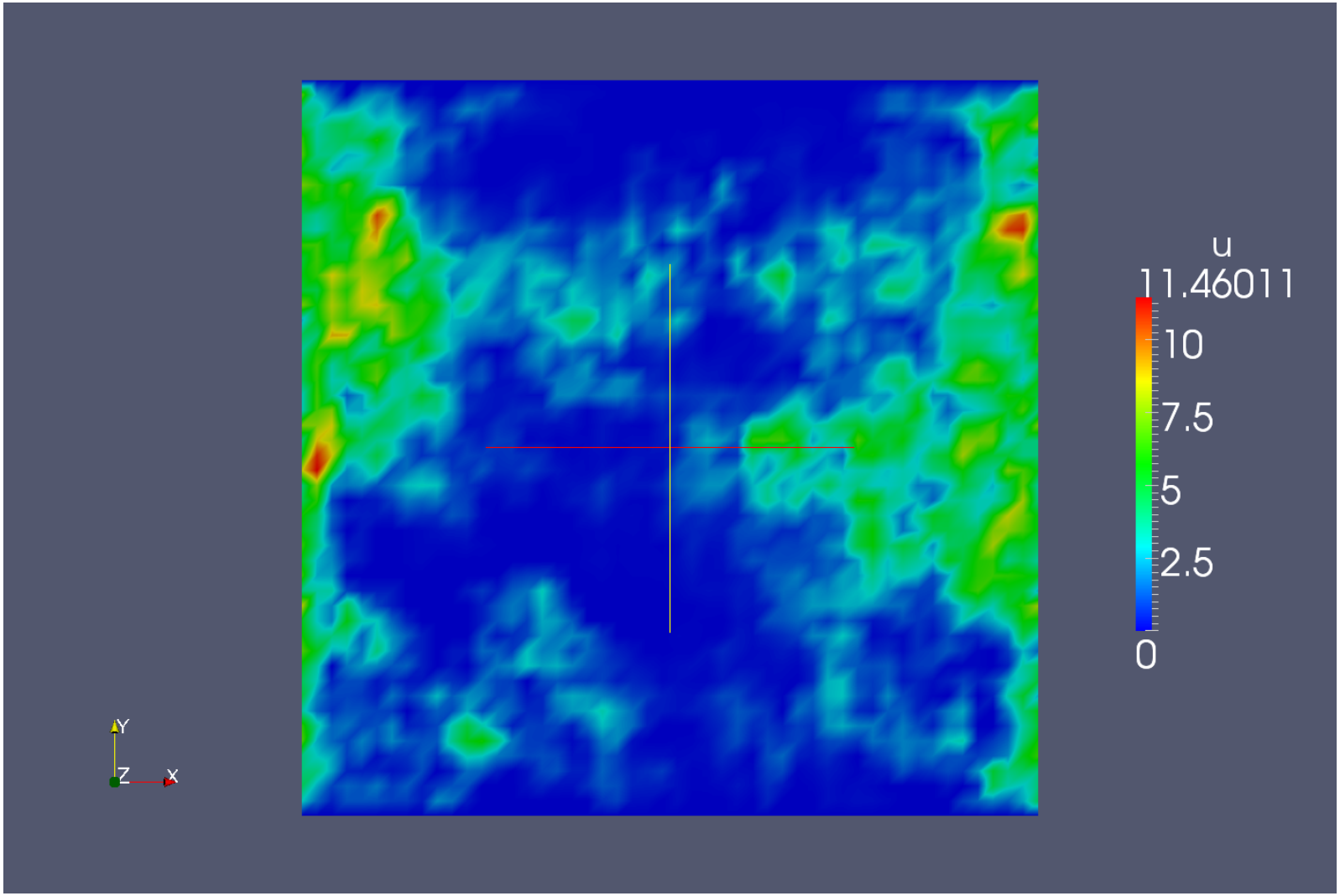


Figure : **Adjoint solution to Poisson Equation.** $1 \times 10^3$ *total histories.*

# Evolution of a Solution: Adjoint Method

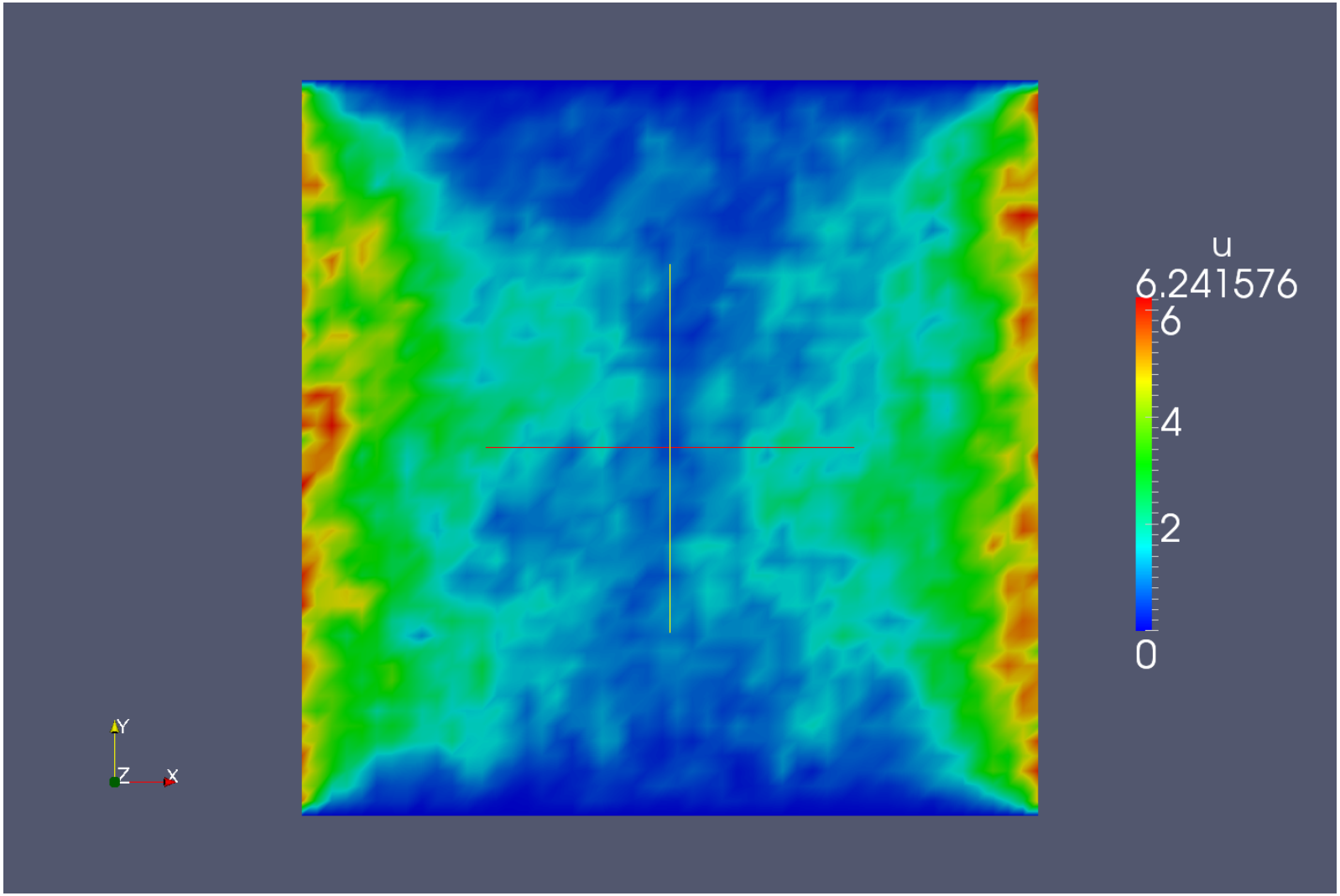


Figure : **Adjoint solution to Poisson Equation.** $1 \times 10^4$ *total histories.*

# Evolution of a Solution: Adjoint Method

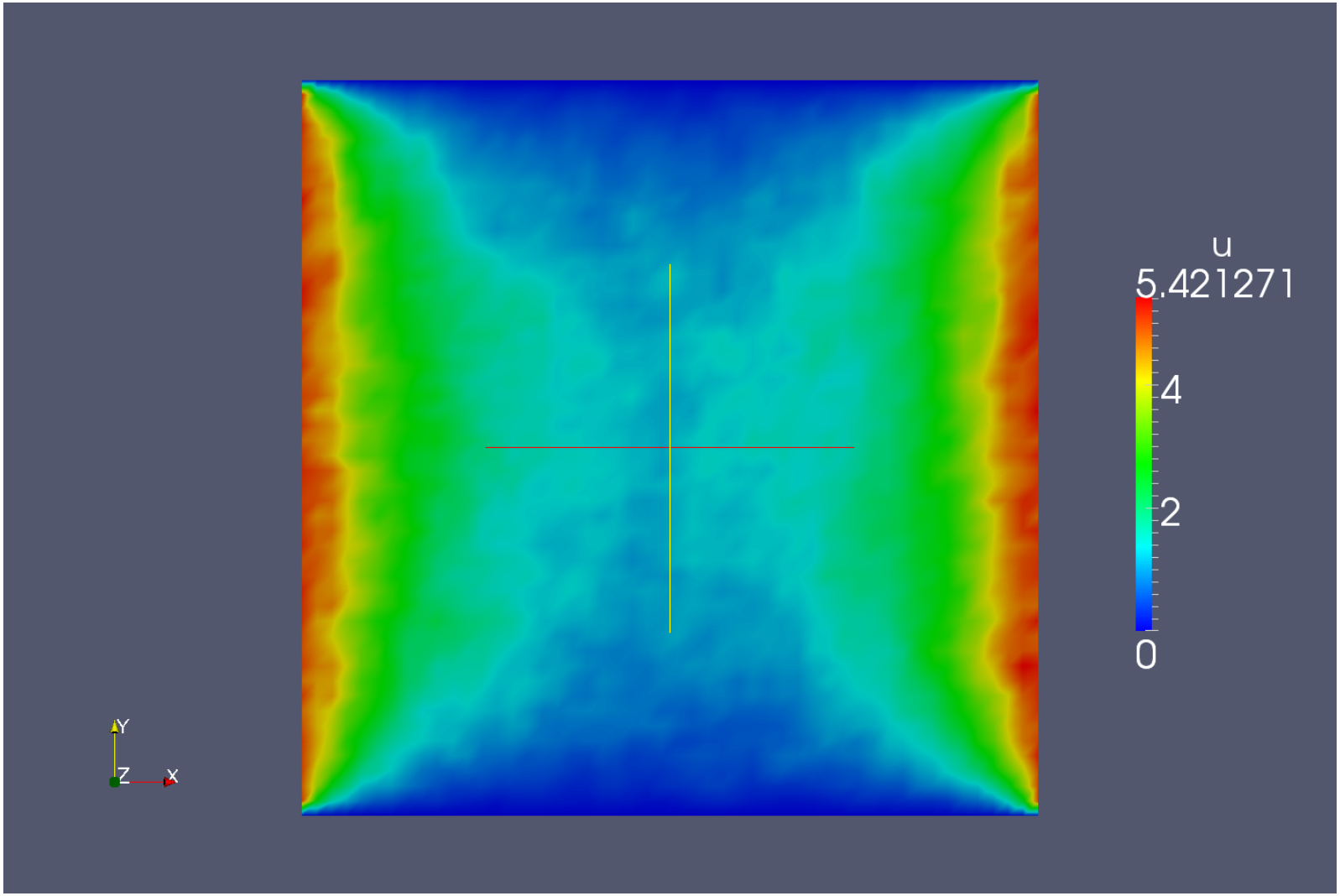


Figure : **Adjoint solution to Poisson Equation.** $1 \times 10^5$ *total histories.*

# Evolution of a Solution: Adjoint Method

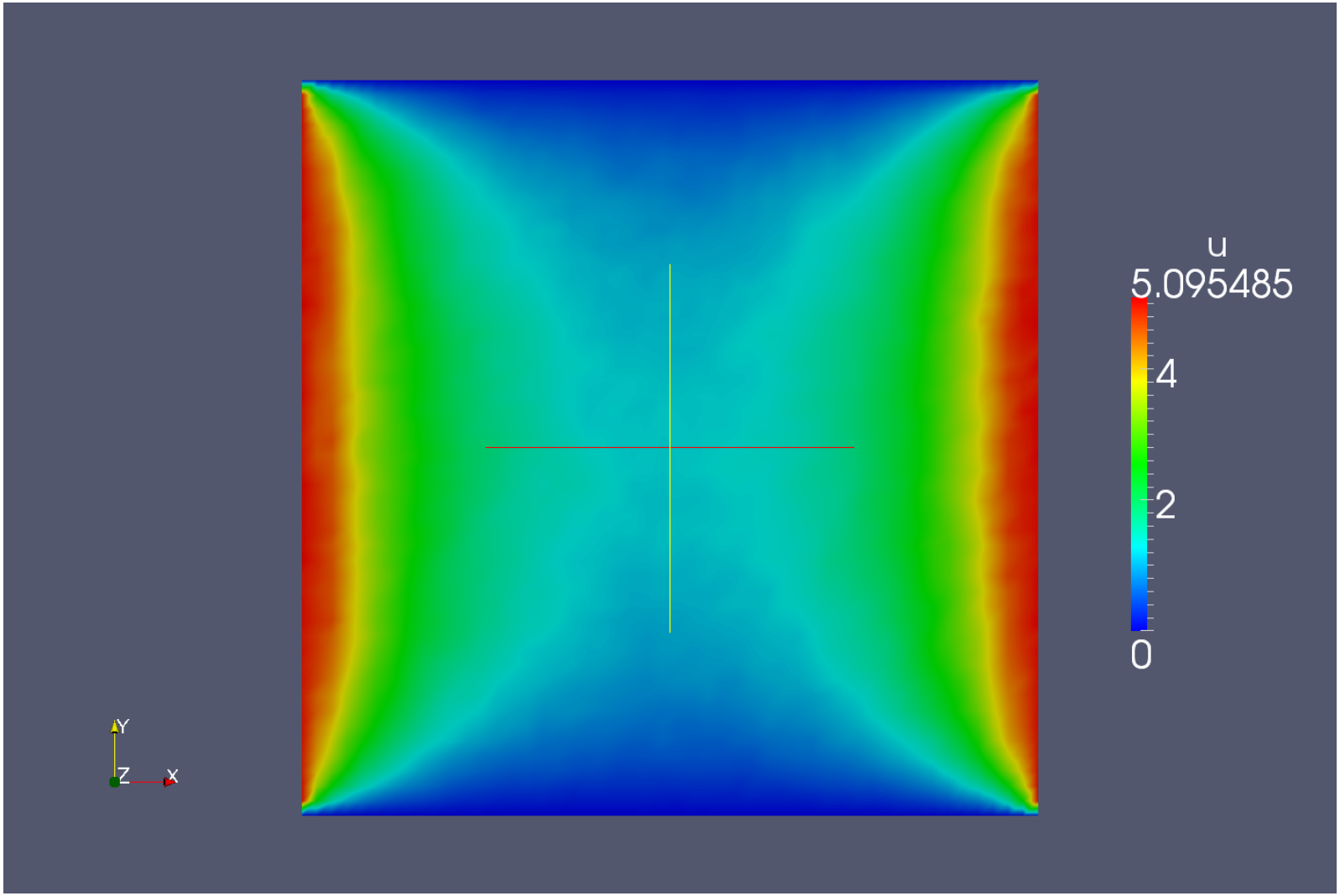


Figure : **Adjoint solution to Poisson Equation.** $1 \times 10^6$ *total histories.*

# Evolution of a Solution: Adjoint Method

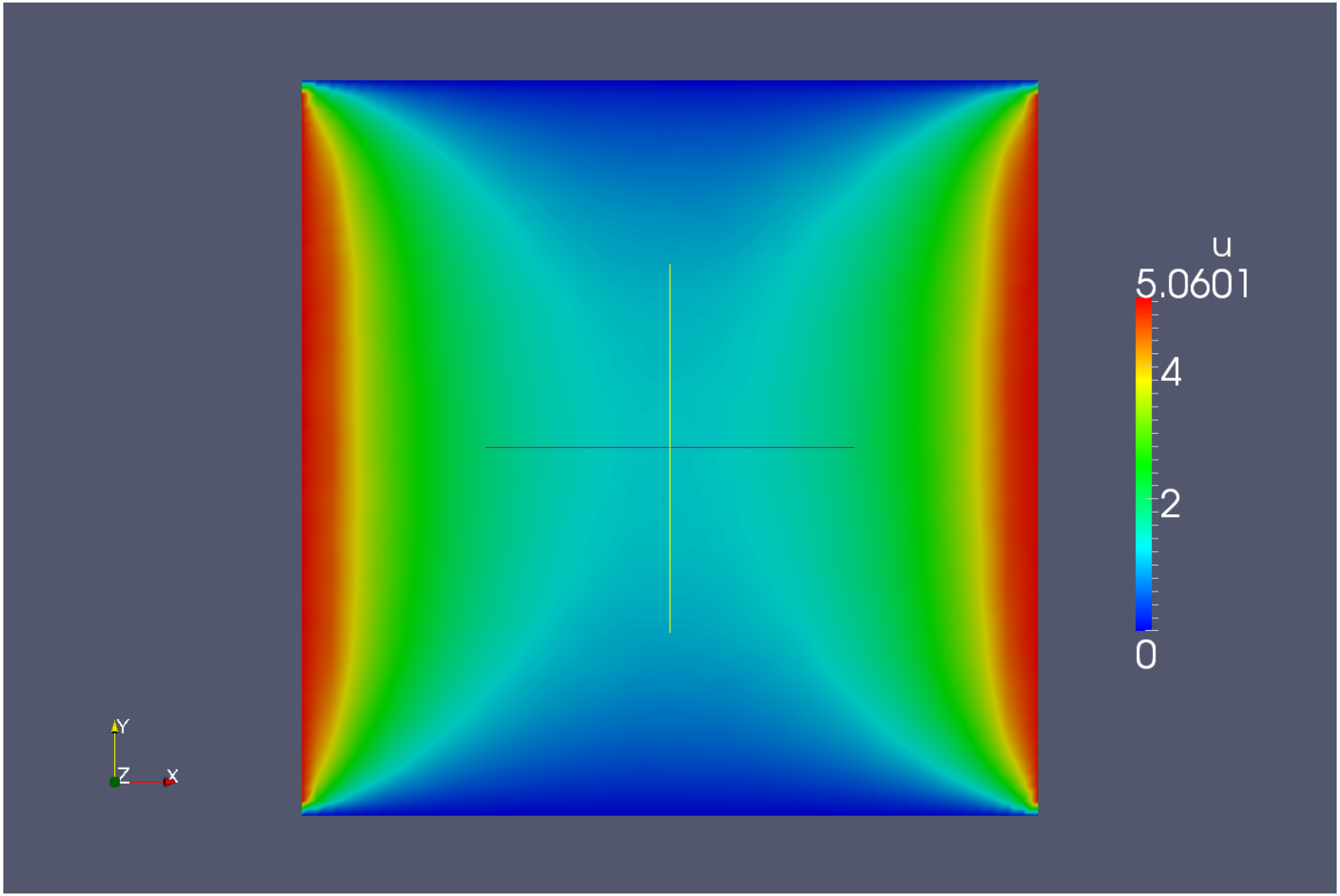


Figure : **Adjoint solution to Poisson Equation.** $1 \times 10^7$ *total histories.*

# Model Problem

Choose a simple homogeneous problem with Dirichlet conditions:

$$\nabla^2 x = 0, \ \mathbf{x}_1 = 0, \quad \mathbf{x}_N = 0$$

Second order finite difference:

$$(\nabla \mathbf{u})_i = \frac{\mathbf{u}_{i-1} - 2\mathbf{u}_i + \mathbf{u}_{i+1}}{h^2}$$

Monte Carlo requires $\rho(\mathbf{H})$ - scale by the diagonal:

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{0}$$

Choose initial guess to be some Fourier mode

$$\mathbf{x}_i^0 = \sin\left(\frac{ik\pi}{N}\right)$$

# Error Analysis

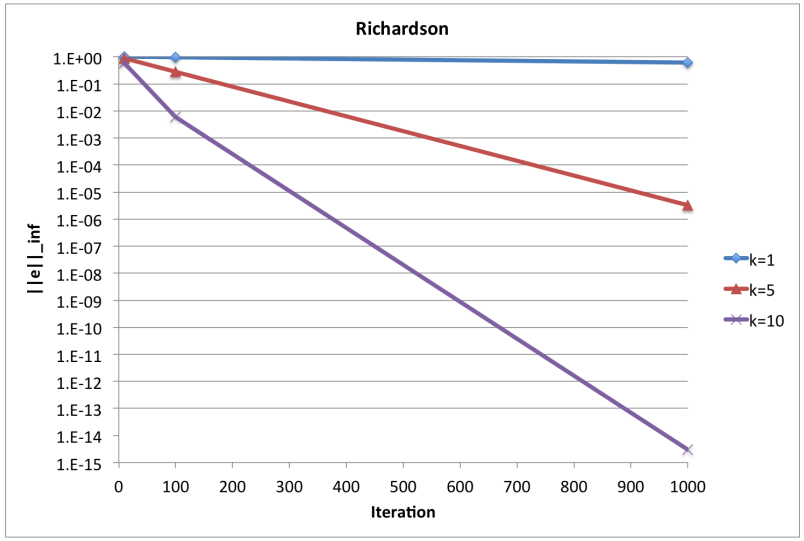


Figure : **Convergence of Richardson's iteration.** *Better for larger wave numbers.*

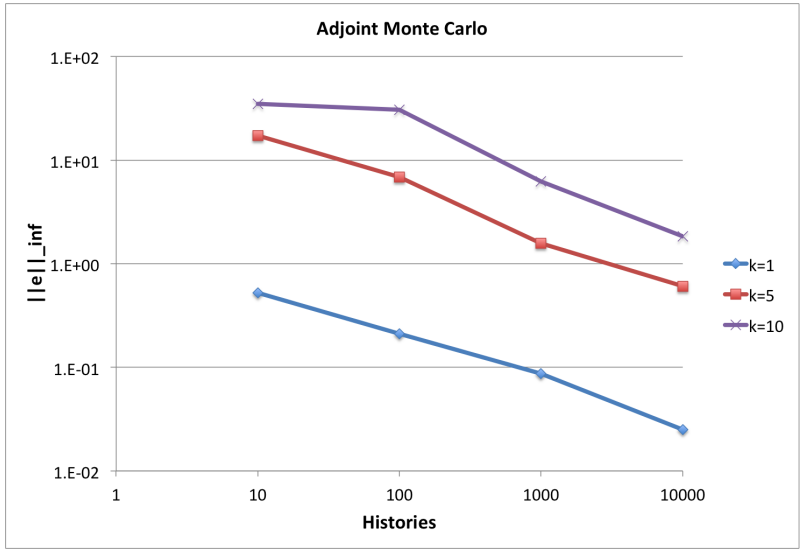


Figure : **Convergence of the adjoint Monte Carlo method.** *Better for smaller wave numbers.*

- A multilevel scheme means larger errors at coarser levels
- More samples required at the coarser levels
- Potential reduction in run time if the coarse level problems are fast

# Error Analysis

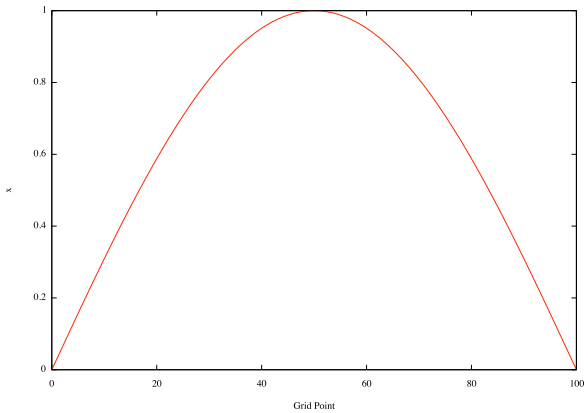


Figure : $k = 1$.

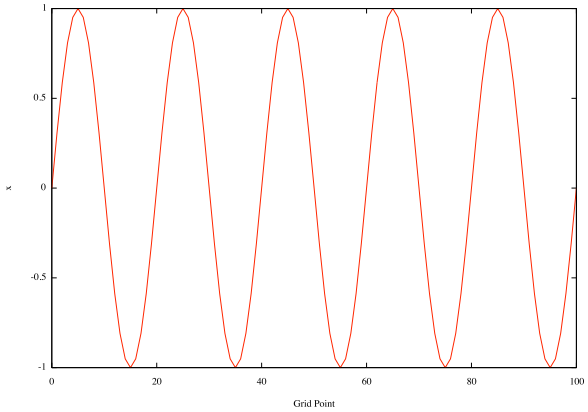


Figure : $k = 10$.

| Wave Number | Time per History (s) |
|---|---|
| 1 | 1 |
| 5 | 0.85 |
| 10 | 0.83 |

Table : **Normalized average time per history.**

- $\sigma(A)$ dictates the characteristics of the Markov chain
- Random walks are shorter for larger $k$
- $N$ dictates the convergence of the Monte Carlo method
- More samples required to resolve fine structures

# Multilevel Monte Carlo Methods

- Formalized for integral equations (Heinrich,2001)
- Expanded for time-dependent finance calculations (Giles,2008)

- Recent work includes Bayesian inference techniques for stochastic PDEs in ground water flow (Cliffe,2011) (Teckentrup,2013)
- Idea is to leverage multigrid ideas to reduce variance

# Multilevel Expectation

Start first with the standard Monte Carlo estimator for the solution vector:

$$\hat{\mathbf{x}} = \frac{1}{N} \sum_{m=1}^{N} x^m$$

Consider $L$ levels with level 0 the finest $L$ the coarsest:

$$E(\mathbf{x}_0) = E(\mathbf{x}_L) + E(\mathbf{x}_{L-1} - \mathbf{x}_L) + E(\mathbf{x}_{L-2} - \mathbf{x}_{L-1}) + \cdots + E(\mathbf{x}_0 - \mathbf{x}_1)$$

Reduce to a sum:

$$\hat{\mathbf{y}}_l = \frac{1}{N_l} \sum_{m=1}^{N_l} (x_l^m - x_{l+1}^m)$$

# Multilevel Expectation

Build a correction estimator for a given level $l$:

$$\hat{\mathbf{y}}_l = \frac{1}{N_l} \sum_{m=1}^{N_l} (x_l^m - x_{l+1}^m)$$

Leaving a final multilevel estimator of:

$$\hat{\mathbf{x}} = \sum_{l=0}^{L} \hat{\mathbf{y}}_l$$

**Critical observation:** $x_l^m$ and $x_{l+1}^m$ must be constructed from the *same* Markov chain

# Constructing Multilevel Estimates

Define a *prolongation operator*, $\mathbf{P}_l$, and a *restriction operator*, $\mathbf{R}_l$, with variational conditions:

$$\mathbf{R} = c\mathbf{P}^T$$

$$\mathbf{A}_{l+1} = \mathbf{R}_l \mathbf{A}_l \mathbf{P}_l$$

Use to build the multilevel estimate:

$$E(\mathbf{x}_l - \mathbf{x}_{l+1}) = \left(\mathbf{I} - \mathbf{P}_l \mathbf{R}_l\right)\hat{\mathbf{x}}_l$$

where $\hat{\mathbf{x}}_l$ is constructed from the standard adjoint estimator at level $l$

Number of samples at each level should be determined from the estimated variance. For simplicity:

$$N_l = 2^{-3(L-l)/2} N$$

**Algorithm 1** Multilevel Monte Carlo Method

1: **for** l = 0...L **do**
2: $\mathbf{P}_l = P(\mathbf{A}_l)$ {Build the prolongation and restriction operators for the $l^{th}$ level.}
3: $\mathbf{R}_l = c\mathbf{P}_l^T$
4: $\mathbf{r}_l = \mathbf{b}_l - \mathbf{A}_l\mathbf{x}_l^0$ {Build the $l^{th}$ level residual.}
5: $\mathbf{d}_l = \hat{\mathbf{A}}_l^{-1}\mathbf{r}_l$ {Solve the $l^{th}$ level problem with adjoint Monte Carlo}
6: **if** l != L **then**
7: $\mathbf{d}_l = (\mathbf{I} - \mathbf{P}_l\mathbf{R}_l)\mathbf{d}_l$ {Apply the multilevel tally}
8: $\mathbf{A}_{l+1} = \mathbf{R}_l\mathbf{A}_l\mathbf{P}_l$ {Construct the next level.}
9: $\mathbf{x}_{l+1}^0 = \mathbf{R}_l\mathbf{x}_l^0$
10: $\mathbf{b}_{l+1} = \mathbf{R}_l\mathbf{b}_l$
11: **end if**
12: **end for**
13: **for** l = L...1 **do**
14: $\mathbf{d}_{l-1} = \mathbf{d}_{l-1} + \mathbf{P}_l\mathbf{d}_l$ {Collapse the tallies to the finest grid}
15: **end for**
16: $\mathbf{x} = \mathbf{x}^0 + \mathbf{d}_0$

# Numerical Experiments

- Solve the model problem on grid size 1024 and $N = 10,000$
- Geometric multigrid operators from Briggs' multigrid tutorial, $M = 2$
- Algebraic multigrid operators from ML, $M \approx 3$

Measure performance with a *figure of merit*:

$$FOM = \frac{1}{||\mathbf{e}||_\infty^2 T}$$

# Geometric Multigrid Results

| Levels | Samples | $\|\|e\|\|_\infty$ | Time (s) | RFOM |
|---|---|---|---|---|
| 1 | 10,000 | 0.022 | 119.1 | 1 |
| 2 | 13,535 | 0.026 | 72.0 | 1.14 |
| 3 | 14,785 | 0.035 | 33.2 | 1.38 |
| 4 | 15,226 | 0.039 | 13.7 | 2.63 |
| 5 | 15,382 | 0.027 | 5.5 | 13.86 |
| 6 | 15,437 | 0.036 | 2.3 | 19.21 |
| 7 | 15,456 | 0.045 | 0.98 | 28.00 |
| 8 | 15,462 | 0.081 | 0.41 | 20.66 |
| 9 | 15,464 | 0.267 | 0.17 | 11.72 |

Table : $k = 1$.

| Levels | Samples | $\|\|e\|\|_\infty$ | Time (s) | RFOM |
|---|---|---|---|---|
| 1 | 10,000 | 0.668 | 101.6 | 1 |
| 2 | 13,535 | 0.381 | 60.0 | 5.21 |
| 3 | 14,785 | 0.396 | 28.0 | 10.34 |
| 4 | 15,226 | 0.599 | 11.9 | 10.64 |
| 5 | 15,382 | 0.638 | 4.7 | 23.60 |
| 6 | 15,437 | 1.052 | 1.8 | 23.15 |
| 7 | 15,456 | 1.070 | 0.67 | 59.16 |
| 8 | 15,462 | 1.180 | 0.23 | 144.10 |
| 9 | 15,464 | 2.130 | 0.10 | 99.95 |

Table : $k = 5$.

| Levels | Samples | $\|\|e\|\|_\infty$ | Time (s) | RFOM |
|---|---|---|---|---|
| 1 | 10,000 | 1.81 | 98.6 | 1 |
| 2 | 13,535 | 1.67 | 59.6 | 1.94 |
| 3 | 14,785 | 2.41 | 27.1 | 2.05 |
| 4 | 15,226 | 3.20 | 11.5 | 2.74 |
| 5 | 15,382 | 1.85 | 4.93 | 19.14 |
| 6 | 15,437 | 3.89 | 1.95 | 10.95 |
| 7 | 15,456 | 3.79 | 0.78 | 28.68 |
| 8 | 15,462 | 7.08 | 0.30 | 21.62 |
| 9 | 15,464 | 11.7 | 0.11 | 21.45 |

Table : $k = 10$.

- There is a limit to the number of levels
- Too few samples on the fine grid drive up the error

# Algebraic Multigrid Results

| Levels | Samples | $\|\|e\|\|_\infty$ | Time (s) | RFOM |
|---|---|---|---|---|
| 1 | 10,000 | 0.022 | 119.1 | 1 |
| 2 | 11,924 | 0.037 | 31.9 | 1.28 |
| 3 | 12,294 | 0.055 | 7.6 | 2.40 |
| 4 | 12,365 | 0.051 | 1.8 | 11.76 |
| 5 | 12,378 | 0.094 | 0.5 | 12.01 |

Table : $k = 1$.

| Levels | Samples | $\|\|e\|\|_\infty$ | Time (s) | RFOM |
|---|---|---|---|---|
| 1 | 10,000 | 0.668 | 101.6 | 1 |
| 2 | 11,924 | 0.834 | 30.8 | 2.12 |
| 3 | 12,294 | 1.140 | 7.5 | 4.63 |
| 4 | 12,365 | 0.975 | 1.6 | 29.09 |
| 5 | 12,378 | 2.240 | 0.4 | 25.10 |

Table : $k = 5$.

| Levels | Samples | $\|\|e\|\|_\infty$ | Time (s) | RFOM |
|---|---|---|---|---|
| 1 | 10,000 | 1.81 | 98.6 | 1 |
| 2 | 11,924 | 2.33 | 30.0 | 1.99 |
| 3 | 12,294 | 3.19 | 7.6 | 4.20 |
| 4 | 12,365 | 3.31 | 1.8 | 16.29 |
| 5 | 12,378 | 9.98 | 0.5 | 7.21 |

Table : $k = 10$.

- Algebraic operators are also effective
- $M \approx 3$ is an ad hoc estimate - need variance estimates instead

# Summary

- Multilevel Monte Carlo methods can be an effective variance reduction technique by reducing the computation time needed to reach a particular error
- More formal analysis of convergence in the context of linear systems is required
- For large problems, density of $A_l$ for coarse levels increases computational complexity of the Monte Carlo - consider non-Galerkin and sparsification approaches
- Future work includes addition of variance estimation to properly select $N_l$

# Acknowledgements

This work sponsored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy.