

# Parallel Monte Carlo Synthetic Acceleration Methods for Discrete Transport Problems

Stuart R. Slattery  
University of Wisconsin - Madison

September 3, 2013





This work was performed under appointment to the Nuclear Regulatory Commission Fellowship program at the University of Wisconsin - Madison Engineering Physics Department



- **Introduction**
- Monte Carlo Synthetic Acceleration Methods
- Application to Neutron Transport
- Application to Fluid Flow
- Parallelization of MCSA
- Summary

- 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



- New algorithms required to leverage new computational resources
  - Tremendous computational resources are required with  $O(1 \times 10^9)$  element meshes and  $O(100,000)+$  cores used today for neutronics and fluid problems (Evans,2010)(Pawlowski,2012)
- Necessary to determine applicability to reactor physics and potential performance improvements before moving forward with other work
- Physics-driven development
  - Research applicability and potential improvements to neutronics and fluid flow
  - Offer solutions or a potential path forward for the observed issues
  - Work to improve iterative and parallel performance



The goal of this work is to improve the iterative performance and parallel scalability of solutions to discrete linear and nonlinear transport problems by researching and developing a new set of domain decomposed Monte Carlo Synthetic Acceleration methods.

- Development of a linear scheme for the  $SP_N$  equations leveraging Monte Carlo Synthetic Acceleration
  - Application to neutron transport
  - Research is required to study MCSA preconditioning
  - Iterative performance is of concern
- Development of a nonlinear scheme for the Navier-Stokes equations leveraging Monte Carlo Synthetic Acceleration
  - Monte Carlo is a more natural fit
  - Application to fluid flow
  - Convergence of the linear model is of concern
- Parallelization of Monte Carlo Synthetic Acceleration
  - Parallel strategies taken from modern reactor physics methods
  - Research is required to explore varying parallel strategies
  - Parallel scalability is of concern



- Introduction
- **Monte Carlo Synthetic Acceleration Methods**
- Application to Neutron Transport
- Application to Fluid Flow
- Parallelization of MCSA
- Summary





- First proposed by J. Von Neumann and S.M. Ulam in the 1940's
- Earliest published reference in 1950
- General lack of published work
- Modern work by Evans and others has yielded new applications

---

Thomas Evans and Scott Mosher, "A Monte Carlo Synthetic Acceleration method for the non-linear, time-dependent diffusion equation", American Nuclear Society - International Conference on Mathematics, Computational Methods and Reactor Physics, 2009.

- Split the linear operator

$$\mathbf{Ax} = \mathbf{b} \quad \rightarrow \quad \mathbf{x} = \mathbf{Hx} + \mathbf{b}$$

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

- 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

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

- Expand the Neumann series

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

- Define a sequence of state transitions

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

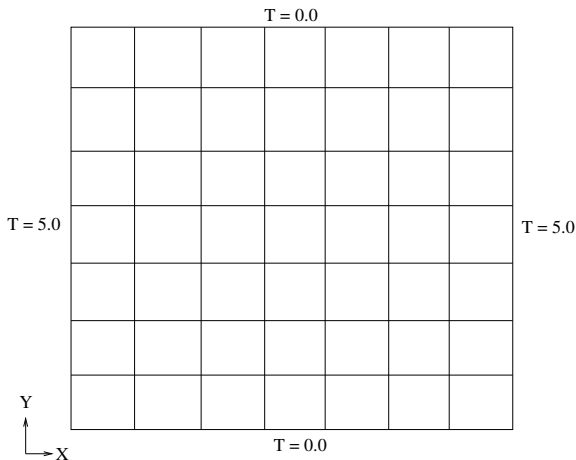
- Use the adjoint Neumann-Ulam decomposition

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

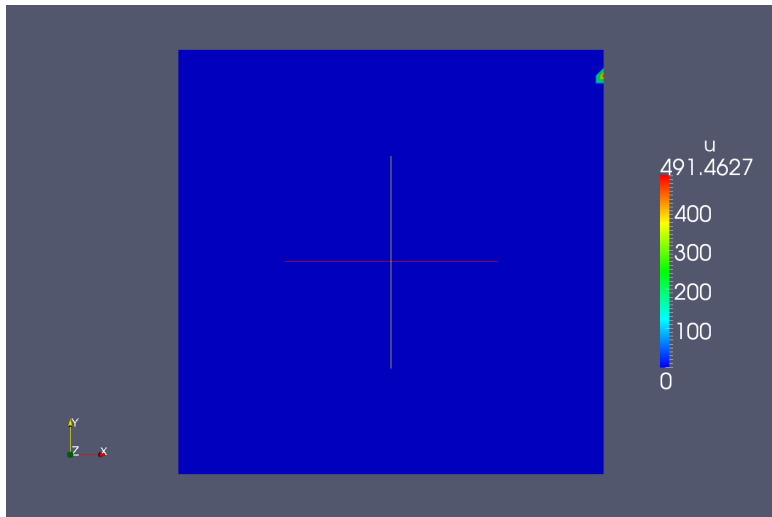
$$p_{ij} = \frac{|h_{ji}|}{\sum_j |h_{ji}|}, \quad w_{ij} = \frac{h_{ji}}{p_{ij}}$$

---

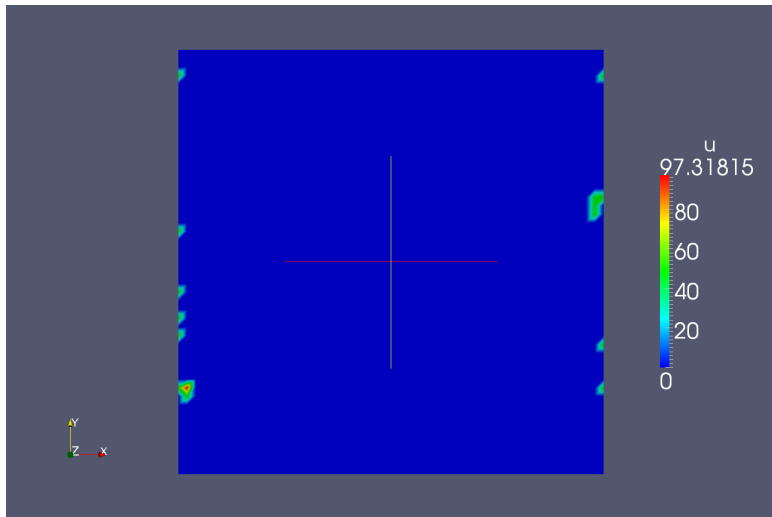
The Hadamard product  $\mathbf{A} = \mathbf{B} \circ \mathbf{C}$  is defined element-wise as  $a_{ij} = b_{ij} c_{ij}$ .



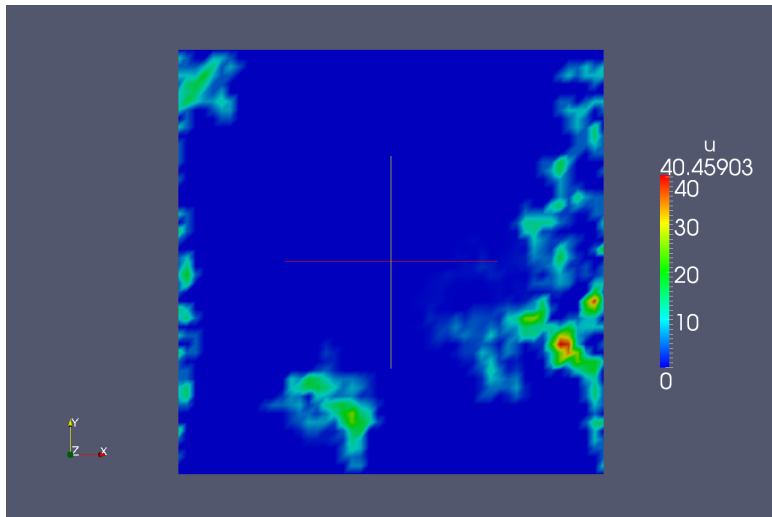
**Figure: Poisson Problem.** *Distributed source of 1.0 in the domain.*



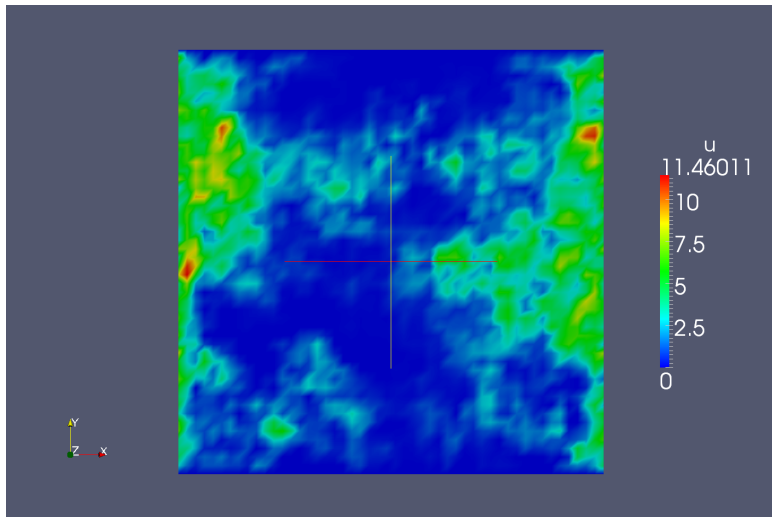
**Figure: Adjoint solution to Poisson Equation.**  $1 \times 10^0$  total histories, 0.286 seconds CPU time.



**Figure:** Adjoint solution to Poisson Equation.  $1 \times 10^1$  total histories, 0.278 seconds CPU time.

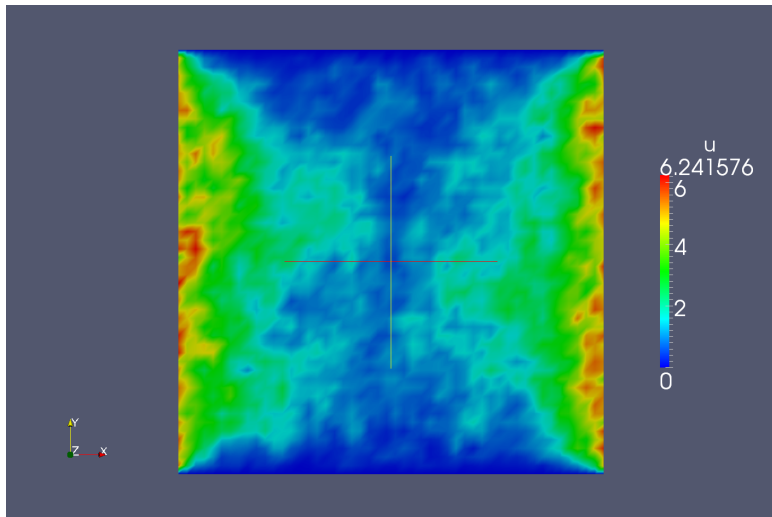


**Figure:** Adjoint solution to Poisson Equation.  $1 \times 10^2$  total histories, 0.275 seconds CPU time.

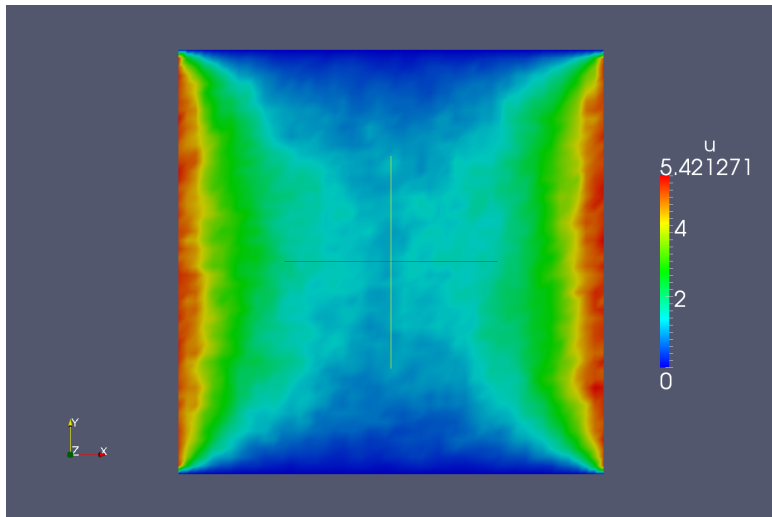


**Figure:** Adjoint solution to Poisson Equation.  $1 \times 10^3$  total histories, 0.291 seconds CPU time.

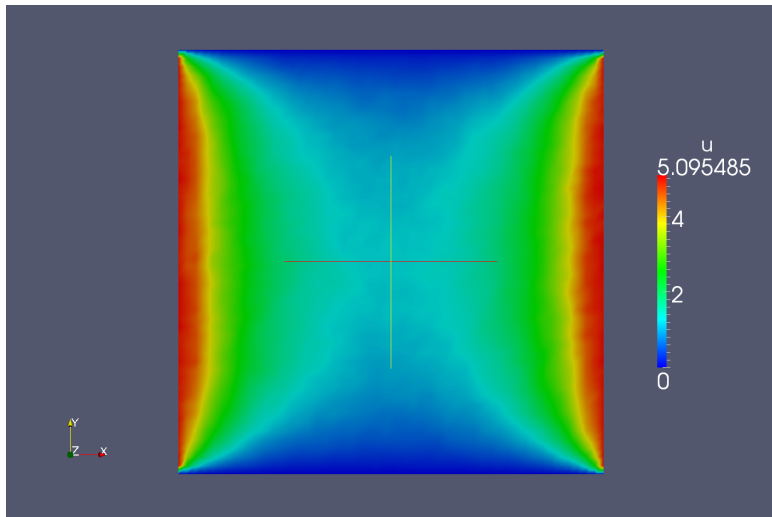




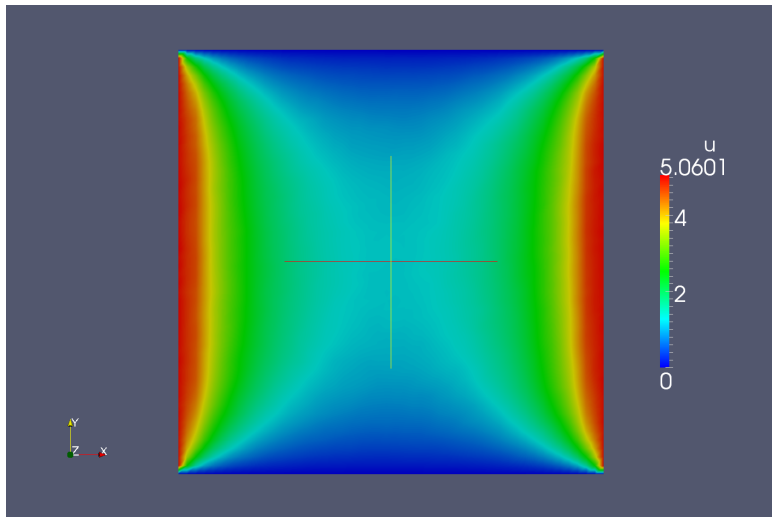
**Figure:** Adjoint solution to Poisson Equation.  $1 \times 10^4$  total histories, 0.428 seconds CPU time.



**Figure:** Adjoint solution to Poisson Equation.  $1 \times 10^5$  total histories, 1.76 seconds CPU time.



**Figure:**      **Adjoint solution to Poisson Equation.**  $1 \times 10^6$  total histories,  
15.1 seconds CPU time.



**Figure:** **Adjoint solution to Poisson Equation.**  $1 \times 10^7$  total histories,  
149 seconds CPU time.

## MCSA Iteration

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

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

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

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

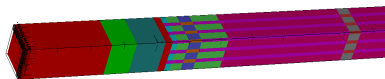
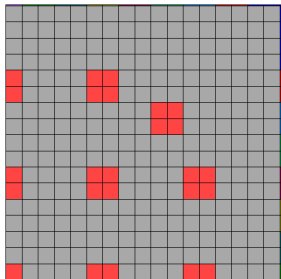
$$\mathbf{x}^{k+1} = \mathbf{x}^{k+1/2} + \delta\mathbf{x}^{k+1/2}$$

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



- 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
- Implemented in C++
- Heavy use of the Trilinos scientific computing libraries
- Open-source BSD 3-clause license
- <https://github.com/sslattery/MCLS>

- Introduction
- Monte Carlo Synthetic Acceleration Methods
- **Application to Neutron Transport**
- Application to Fluid Flow
- Parallelization of MCSA
- Summary



- CASL Problem 3:  $17 \times 17$  quarter symmetry HZP LWR fuel assembly
- Multigroup  $SP_N$  discretization
- MCLS leveraged by the Exnihilo code base (ORNL)
- Emphasize algorithm development for iterative performance

Parameter	Value
Power Level	0 MW
Inlet Temperature	326.85C
Fuel Temperature	600C
Boron Concentration	1300 ppm
Moderator Density	0.743 g/cc
Helium Density	$1.79 \times 10^{-4}$ g/cc
Zirconium Density	6.56 g/cc
Stainless Steel Density	8.0 g/cc
Inconel Density	8.19 g/cc
UO2 Density	10.257 g/cc
Fuel Pin Radius (w/o clad)	0.4096 cm



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

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

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

---

## Algorithm 1 Power Iteration MCSA Scheme

---

$k_0$  = initial guess

$\Phi_0$  = initial guess

$n = 0$

**while**  $\left| \frac{k^n - k^{n-1}}{k^n} \right| < \epsilon$  **do**

$\mathbf{M}\Phi^{n+1} = \frac{1}{k^n} \mathbf{F}\Phi^n$  {Solve for the new flux state with MCSA}

$$k^{n+1} = k^n \frac{\int \mathbf{F}\Phi^{n+1} d\mathbf{r}}{\int \mathbf{F}\Phi^n d\mathbf{r}}$$

$n = n + 1$

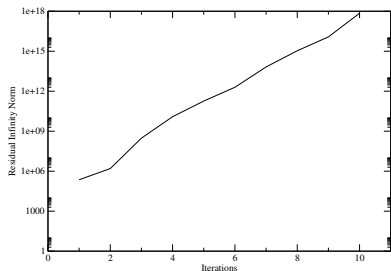
**end while**

---

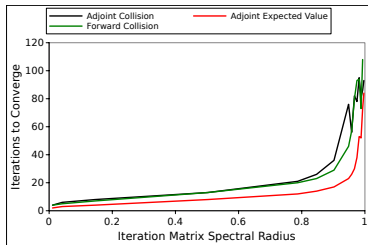
- Inject MCSA as the solver at each eigenvalue operation
- Transport operator is static - one time cost for MCSA setup
- Current Exnihilo implementation gives the full operator

		$SP_N$ Order			
		1	3	5	7
$P_N$ Order	0	0.0647	0.1275	0.1449	0.1514
	1	0.0686	0.1338	0.1484	0.1547
	3	0.0687	0.1399	0.1582	0.1625
	5	0.0692	0.1399	0.1582	0.1657
	7	0.0678	0.1393	0.1624	0.166

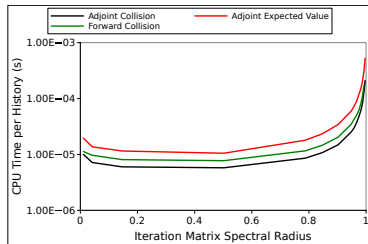
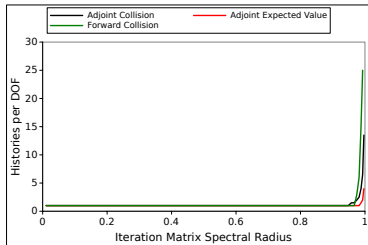
**Table:** Spectral radius results for the block Jacobi preconditioned iteration matrix with 10 energy groups and full downscatter for sample problem.



- Rapidly divergent results
- Light water moderator creates a lot of scattering and  $\rho(\mathbf{H}) \approx 1$
- Convergence not achieved with 50 histories per DOF and 90 minutes compute time



- As  $\rho(\mathbf{H}) \rightarrow 1$  terrible things happen...
- A more robust set of preconditioners is required for the  $SP_N$  equations



$$\mathbf{M}_L^{-1} \mathbf{A} \mathbf{M}_R^{-1} \mathbf{M}_R \mathbf{x} = \mathbf{M}_L^{-1} \mathbf{b} \quad \rightarrow \quad \mathbf{M}_L^{-1} \mathbf{A} \mathbf{M}_R^{-1} \mathbf{u} = \mathbf{M}_L^{-1} \mathbf{b}$$

$$\mathbf{x} = \mathbf{M}_R^{-1} \mathbf{u}$$

## Left/Right Preconditioned MCSA Iteration

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

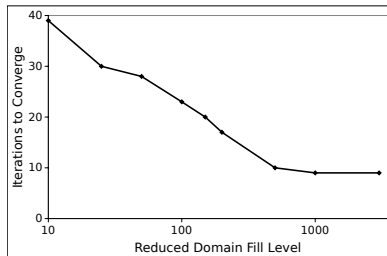
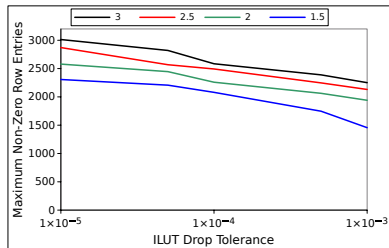
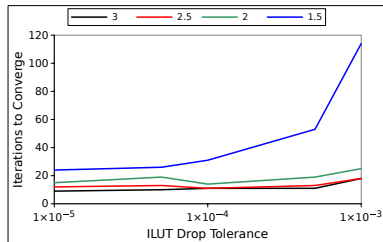
$$\mathbf{u}^{k+1/2} = \mathbf{u}^k + \mathbf{r}^k$$

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

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

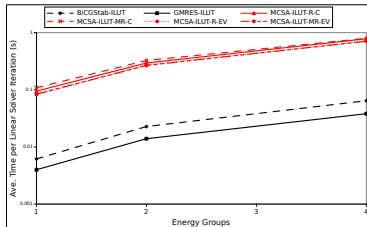
$$\mathbf{u}^{k+1} = \mathbf{u}^{k+1/2} + \delta \mathbf{u}^{k+1/2}$$

- Factor transport operator into upper and lower triangular parts
  - $\mathbf{R} = \mathbf{LU} - \mathbf{M}$
- Control factorization content with level-of-fill and drop tolerance
- Use the reduced domain approximation to recover sparsity



Solver	1 Group	2 Groups	4 Groups
BiCGStab-ILUT	11.6	11.6	12.4
GMRES-ILUT	18.1	17.9	18.9
MCSA-ILUT-R-C	14.6	15.4	17.6
MCSA-ILUT-MR-C	16.0	17.1	23.7
MCSA-ILUT-R-EV	18.3	19.4	16.8
MCSA-ILUT-MR-EV	19.6	22.4	17.5
Richardson-ILUT	60.9	60.4	63.4

**Table:** Average number of linear solver iterations per eigenvalue iteration.



- Comparison to Trilinos Aztec Krylov solvers with ILUT
- MCLS generates the same k-eigenvalue and neutron flux in all groups and spatial locations
- MCSA converged in fewer iterations than GMRES, more iterations than BiCGStab
- Explicit preconditioning strategy destroys sparsity and elevates CPU times (Ifpack ILUT)
- Spectral radius and memory limitations combine to prevent solutions at finer discretizations

- Introduction
- Monte Carlo Synthetic Acceleration Methods
- Application to Neutron Transport
- **Application to Fluid Flow**
- Parallelization of MCSA
- Summary



- Sequence of Navier-Stokes benchmarks
  - Thermal convection cavity problem (De Vahl Davis, 1983)
  - Lid driven cavity problem (Ghia et al., 1982)
  - Backward-Facing step problem (Gartling, 1990)
- Tuning benchmark parameters varies the strength of nonlinearities

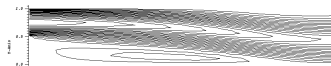
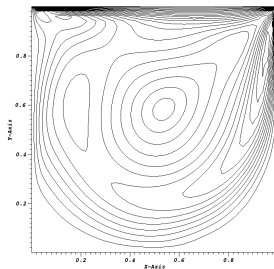
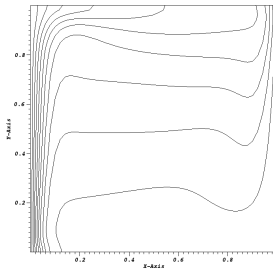
$$\rho \mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot \mathbf{T} - \rho \mathbf{g} = \mathbf{0}$$

$$\nabla \cdot \mathbf{u} = 0$$

$$\rho C_p \mathbf{u} \cdot \nabla T + \nabla \cdot \mathbf{q} = 0$$

$$\mathbf{T} = -P\mathbf{I} + \mu[\nabla \mathbf{u} + \nabla \mathbf{u}^T]$$

$$\mathbf{q} = -k \nabla T$$





- Need a nonlinear solution scheme for the Navier-Stokes equations
- Significant research on Newton methods since the 1980's
- Newton methods often leverage Krylov solvers
  - Robust implementations
  - No operator required
- Monte Carlo methods need the full operator
- Automatic construction of the linear model is available
  - Operator overloading for nonlinear residual differentiation
  - Ideal for Monte Carlo
  - Similar framework properties to matrix-free methods

- Seek solutions of the general nonlinear problem

$$\mathbf{F}(\mathbf{u}) = \mathbf{0}$$

$$\mathbf{u} \in \mathbb{R}^n, \mathbf{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$$

- Interpret the exact solution  $\mathbf{u}$  to be the roots of  $\mathbf{F}(\mathbf{u})$

$$\mathbf{F}(\mathbf{u}^{k+1}) = \mathbf{F}(\mathbf{u}^k) + \mathbf{F}'(\mathbf{u}^k)(\mathbf{u}^{k+1} - \mathbf{u}^k) + \frac{\mathbf{F}''(\mathbf{u}^k)}{2}(\mathbf{u}^{k+1} - \mathbf{u}^k)^2 + \dots$$

- Form Newton's method

$$\mathbf{J}(\mathbf{u})\delta\mathbf{u}^k = -\mathbf{F}(\mathbf{u}^k)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \delta\mathbf{u}^k$$

## Forward-Automated Newton-MCSA

---

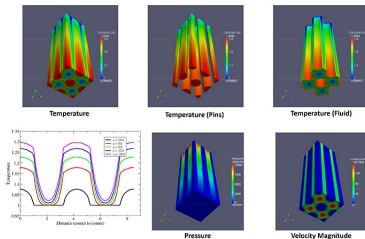
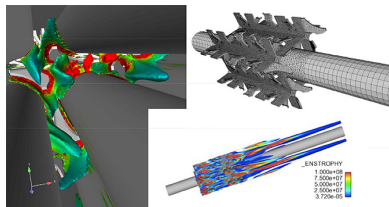
### Algorithm 2 FANM

---

```
1:  $k := 0$ 
2: while  $\|\mathbf{F}(\mathbf{u}^k)\| > \epsilon \|\mathbf{F}(\mathbf{u}^0)\|$  do
3:    $\mathbf{J}(\mathbf{u}^k) \leftarrow AD(\mathbf{F}(\mathbf{u}^k))$  {Automatic differentiation}
4:    $\mathbf{J}(\mathbf{u}^k)\delta\mathbf{u}^k = -\mathbf{F}(\mathbf{u}^k)$  {Solve for the Newton correction with MCSA}
5:    $\mathbf{u}^{k+1} \leftarrow \mathbf{u}^k + \delta\mathbf{u}^k$ 
6:    $k \leftarrow k + 1$ 
7: end while
```

---

- Robustness of Newton's method (inexact)
- Accuracy and convenience of FAD
- Potential parallelism, and resiliency benefits of MCSA
- Requires only nonlinear function evaluations
- Can utilize globalization and forcing term selection methods



- Drekar is a production fluid base using Newton methods with FAD
- MCLS incorporated into Drekar nonlinear scheme to implement FANM
- Newton-Krylov method with Aztec GMRES used for benchmark comparisons
- All problems preconditioned with explicit scheme using algebraic multigrid (ML) and leveraged some kind of globalization (e.g. backtracking)

*Images source: [www.casl.gov](http://www.casl.gov)*

Benchmark	NK	FANM	NR
Convection, $Ra=1 \times 10^3$	5	5	5
Convection, $Ra=1 \times 10^4$	7	7	7
Convection, $Ra=1 \times 10^5$	9	10	9
Convection, $Ra=1 \times 10^6$	11	11	11
Lid Driven, $Re=100$	6	6	7
Lid Driven, $Re=300$	9	9	9
Lid Driven, $Re=500$	11	11	11
Lid Driven, $Re=700$	14	10	12
Backward Step, $Re=200$	10	9	10
Backward Step, $Re=300$	15	14	15
Backward Step, $Re=400$	10	10	10
Backward Step, $Re=500$	19	20	21

**Table: Navier-Stokes benchmark comparison for nonlinear iterations.** Over all benchmarks, FANM performed better in terms of nonlinear iterations for 1 more case than the Newton-Krylov method.

Benchmark	GMRES	MCSA	Richardson
Convection, $Ra=1 \times 10^3$	32	18	38
Convection, $Ra=1 \times 10^4$	23	17	34
Convection, $Ra=1 \times 10^5$	25	20	34
Convection, $Ra=1 \times 10^6$	39	25	48
Lid Driven, $Re=100$	27	42	151
Lid Driven, $Re=300$	35	52	133
Lid Driven, $Re=500$	41	56	154
Lid Driven, $Re=700$	21	14	32
Backward Step, $Re=200$	24	13	21
Backward Step, $Re=300$	23	17	21
Backward Step, $Re=400$	18	12	14
Backward Step, $Re=500$	30	52	98

**Table: Navier-Stokes benchmark comparison for total linear solver iterations.** *Over all benchmarks, FANM performed better in terms of linear solver iterations for twice as many cases as the Newton-Krylov method.*

Benchmark	NK Speedup
Convection, $Ra=1 \times 10^3$	338
Convection, $Ra=1 \times 10^4$	336
Convection, $Ra=1 \times 10^5$	346
Convection, $Ra=1 \times 10^6$	465
Lid Driven, $Re=100$	299
Lid Driven, $Re=300$	322
Lid Driven, $Re=500$	288
Lid Driven, $Re=700$	488
Backward Step, $Re=200$	400
Backward Step, $Re=300$	593
Backward Step, $Re=400$	825
Backward Step, $Re=500$	1057

**Table: Newton-Krylov speedup over FANM.** *For all benchmarks the explicit MCSA preconditioning strategy caused significantly larger CPU times for FANM when compared to the Newton-Krylov solutions.*



- Introduction
- Monte Carlo Synthetic Acceleration Methods
- Application to Neutron Transport
- Application to Fluid Flow
- **Parallelization of MCSA**
- Summary



- No literature observed for parallel Neumann-Ulam solvers beyond history-level parallelism
- Numerous references for modern parallel Monte Carlo methods in reactor physics
- Build a strategy for applying modern methods to the Neumann-Ulam method
- MCSA iteration-level parallelism comes from parallel matrix/vector operations

- Domain decomposition determined by the input system

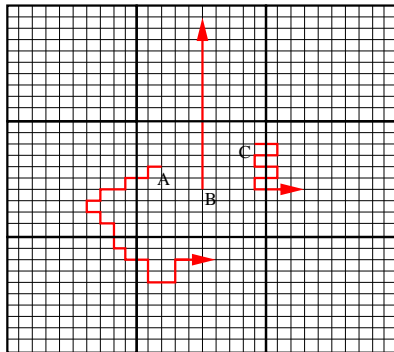
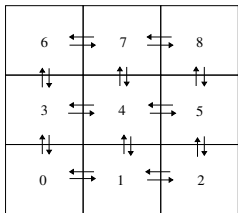


Figure: Domain decomposition example illustrating how domain-to-domain transport creates communication costs.

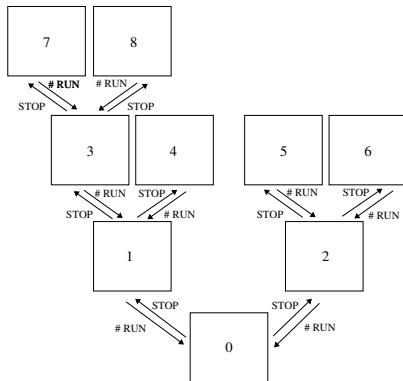
# Asynchronous Monte Carlo Transport Kernel

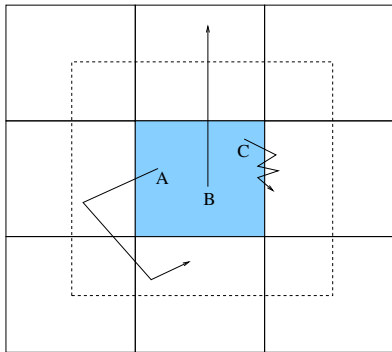


- Developed by Brunner and Brantley in 2009
- Asynchronous nearest neighbor communication of histories
- Binary asynchronous communication tree for completing transport



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





**Figure:** *Overlapping domain example illustrating how domain overlap can reduce communication costs.*

- Developed by Wagner and colleagues in 2010
- Each set contains the full domain
- Multiple sets replicate the domain
- Domains overlap within a set
- Redundancy for resiliency (and useful work)

---

Wagner et. al., "Hybrid and parallel domain-decomposition methods development to enable Monte Carlo for reactor analysis", Joint International Conference on Supercomputing in Nuclear Applications and Monte Carl (SNA+MC 2010), 2010.

# Multiple-Set Overlapping-Domain Decomposition

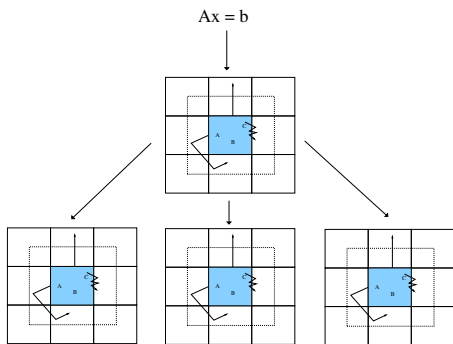


Figure: *MSOD construction.*

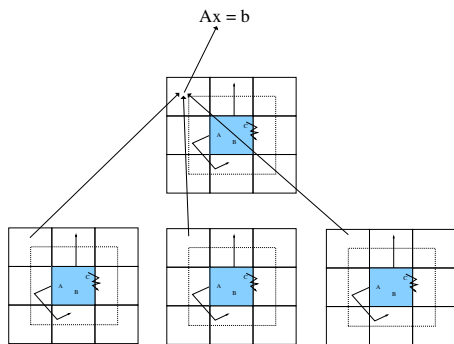


Figure: *MSOD tally reduction.*



- Simple 2D neutron diffusion problem for control - spectral radius is maintained as global problem size grows
- Comparison to Trilinos Belos Krylov solvers with Jacobi preconditioning - conjugate gradient and GMRES
- Strong scaling - Global size fixed at  $1.6E7$  DOFs
- Weak scaling - Local size fixed at  $4.0E4$  DOFs
- Calculations performed on the Titan Cray XK7 machine at ORNL (MPI only)
- Limited MCLS arithmetic optimization artificially inflates efficiencies

# Strong Scaling Results

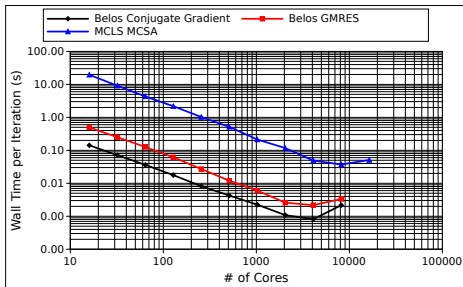


Figure: Wall time.

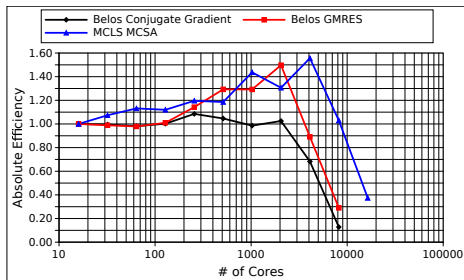
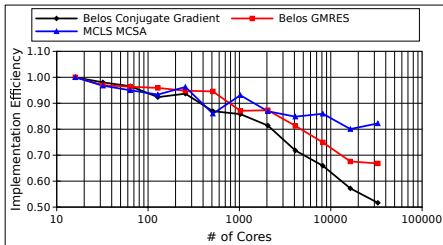
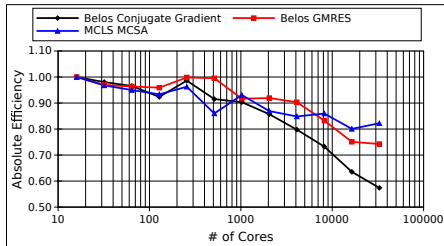
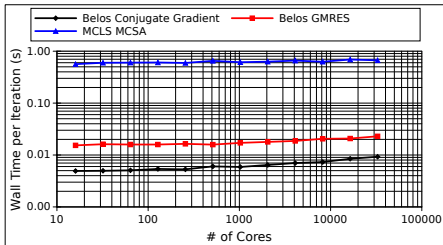


Figure: Absolute efficiency.

- MCLS is an order of magnitude slower arithmetically.
- Super-linear speed-up from memory thrashing in base case.
- CG demonstrates poor scaling due to the cheaper iteration sequence



# Weak Scaling Results



- Krylov implementation efficiencies drop with reduced iteration count
- MCSA maintained constant number of iterations

# Strong Scaling Results with Multiple Sets



**Splitting:** same global number of histories as the single set

**Replicating:** set-multiple of single set problem global histories

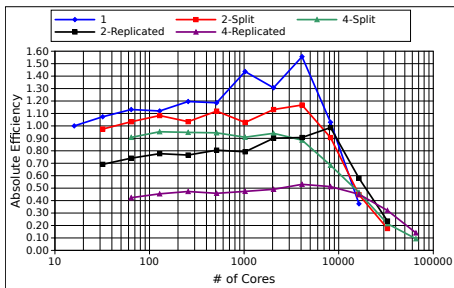


Figure: Absolute efficiency relative to 16-core 1-set base case.

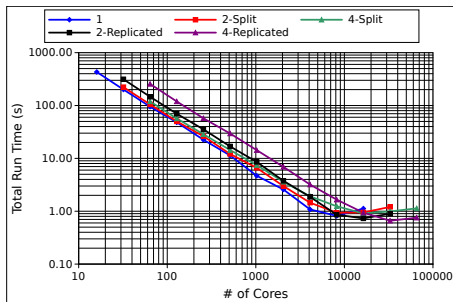


Figure: Wall time

# Weak Scaling Results with Multiple Sets



- Need to consider adding sets is a strong scaling exercise
- Modify the weak scaling efficiency computation to account for these extra resources
- Superposition of Monte Carlo results enhances time to solution!

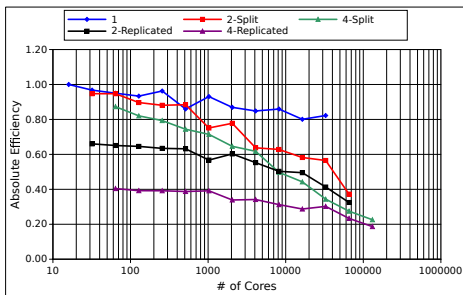


Figure: Absolute efficiency relative to 16-core 1-set base case.

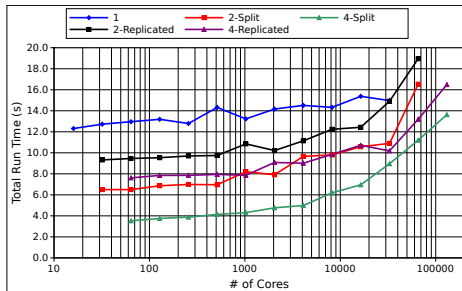
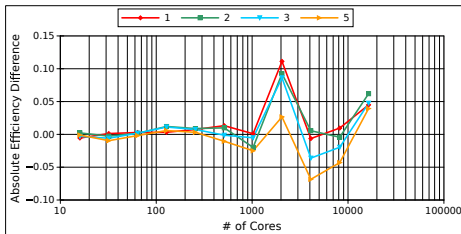


Figure: Wall time.

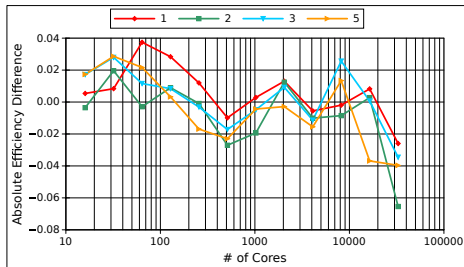
# Scaling Results with Overlap



- Overlap values selected based on average 'diffusion length' of a history in the system of 2.6 discrete states
- Overlap eliminates communication in the Monte Carlo sequence but simply differs it to an overlapping tally vector reduction



**Figure:** Strong scaling efficiency difference compared to the 0 overlap case.



**Figure:** Weak scaling efficiency difference compared to the 0 overlap case.

# MCSA as a Stochastic Additive Schwarz Method



- No domain-to-domain communication in Monte Carlo sequence
- Fixed point iteration acts as a smoother
- Observed to converge in the same number of iterations
- Can add overlap to preserve iterative performance for more ill-conditioned problems

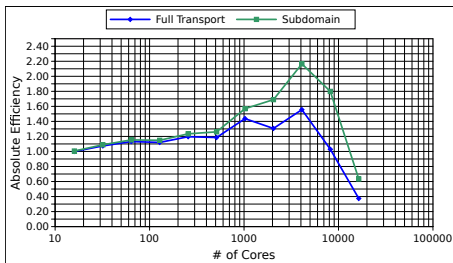


Figure: Strong scaling absolute efficiency relative to full transport base case.

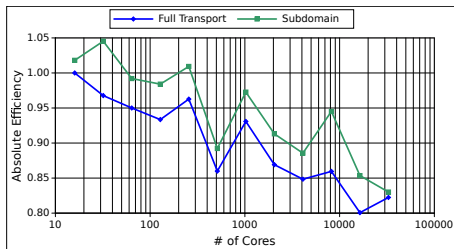


Figure: Weak scaling absolute efficiency relative to full transport base case.

- Introduction
- Monte Carlo Synthetic Acceleration Methods
- Application to Neutron Transport
- Application to Fluid Flow
- Parallelization of MCSA
- **Summary**



- MCSA has been incorporated into the Exnihilo neutronics production code base developed at Oak Ridge National Laboratory
- MCSA can solve the asymmetric system generated by the  $SP_N$  equations
- Light water reactor problems are difficult to solve with MCSA as they have large spectral radii due to the neutron scattering in the moderator
- Advanced algebraic preconditioning strategies were applied to the  $SP_N$  equations to obtain convergence with ILUT chosen for subsequent investigations
- MCSA with the reduced domain approximation was observed to converge in fewer iterations per eigenvalue iteration than GMRES for the fuel assembly criticality problem and more than Bi-CGStab using the same preconditioning



- Forward-Automated Newton-MCSA (FANM) has been developed
- The FANM method has been incorporated into the Drekar multiphysics production code base developed at Sandia National Laboratories
- The FANM method has been verified to produce the same solutions as a production Newton-Krylov method for three difficult benchmark problems for the Navier-Stokes equations in different flow regimes and geometries
- The FANM method has better iterative performance than the Newton-Krylov method for convection dominated problems, converging in fewer linear solver iterations with the same preconditioning for high and low Rayleigh numbers
- The spectral radius convergence restriction on MCSA was observed to be a significant hindrance by preventing solutions to forced flow problems at high Reynolds numbers





- The multiple-set overlapping-domain (MSOD) parallel algorithm for domain decomposed particle transport has been adapted to parallelize MCSA
- MCSA scales favorably compared to production Krylov methods for both strong and weak scaling cases
- Overlap in small quantities can provide parallel efficiency boosts of a few percent in strong scaling cases but is ineffective in weak scaling cases
- Multiple sets offers a means to reduce time to solution by solving multiple copies of the original problem and combining the solutions using superposition
- MCSA is most efficiently used in parallel as a stochastic realization of an additive Schwarz method

- Shortcomings observed on real problems
  - Significant optimization required to determine production feasibility and true scalability
  - Explicit algebraic preconditioning methods not sufficient
  - Spectral radius limitation is severe
- Performance improvements
  - Random walk optimizations
  - Multiple set reduction analysis
  - FANM forcing term and MCSA history relationships
- Preconditioning improvements
  - Variance reduction based strategy
  - Reduced order physics/PDE models for acceleration
- Breaking away from  $\rho(\mathbf{H}) < 1$ 
  - Monte Carlo methods of the second degree

- ① S.R. Slattery, T.M. Evans, P.P.H. Wilson, **A Multiple-Set Overlapping-Domain Decomposed Monte Carlo Synthetic Acceleration Method for Linear Systems**, *Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013 (SNA+MC 2013), Paris, France, October 27-31, 2013. Accepted for oral presentation.*
- ② S.R. Slattery, T.M. Evans, P.P.H. Wilson, **A Spectral Analysis of the Domain Decomposed Monte Carlo Method for Linear Systems**, *International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013), American Nuclear Society, Sun Valley, ID, May 5-9, 2013.*
- ③ T.M. Evans, S.W. Mosher, S.R. Slattery, S.P. Hamilton, **A Monte Carlo Synthetic-Acceleration Method for Solving the Thermal Radiation Diffusion Equation**, *Journal of Computational Physics, Submitted.*



- Paul Wilson
- Tom Evans
- Roger Pawlowski
- CASL, ORNL, Sandia, OLCF