

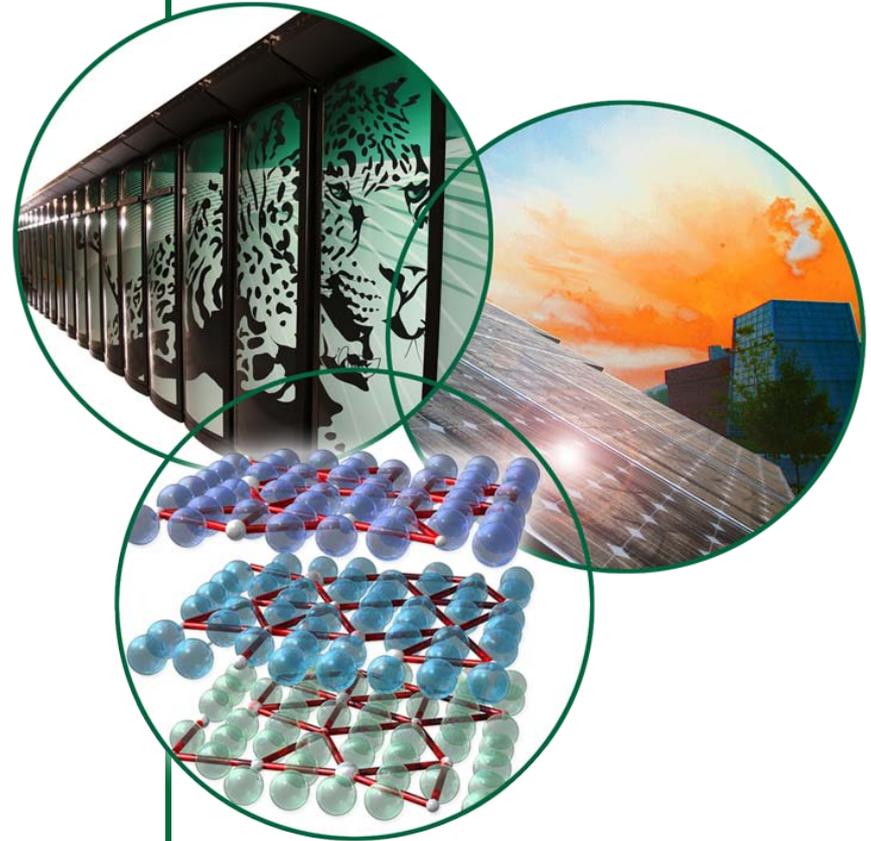
Parallel Algorithms for Fixed-Source and Eigenvalue Problems

Thomas M. Evans

Greg Davidson

Scott Mosher

NSTD Seminar, May 27, 2010



Collaborators

NSTD Staff

- Kevin Clarno
- Douglas Peplow

Students

- Rachel Slaybaugh (Wisconsin)
- Josh Hykes (North Carolina State)
- Alyssa Stafford (Texas A&M)
- Bruno Turcksin (Texas A&M)
- Ahmad Ibrahim (Wisconsin)

OLCF Support

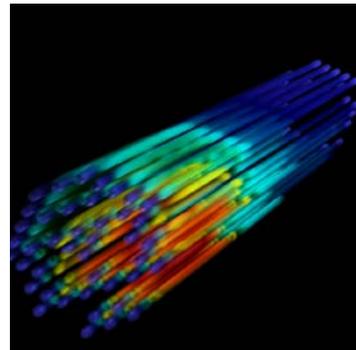
- Dave Pugmire
- Sean Ahern
- Wayne Joubert
- Chris Baker

Outline

- Denovo Capabilities Overview
- Transport Solvers
- KBA Sweep Algorithm
- Multilevel Parallel Decompositions
- First-collision Parallel Algorithms
- Summary

Denovo Capabilities

- State of the art transport methods
 - 3D, non-uniform, regular grid SN
 - Multigroup energy, anisotropic Pn scattering
 - Forward/Adjoint
 - Fixed-source/ k -eigenvalue
 - 6 spatial discretization algorithms
 - Linear and Trilinear discontinuous FE, step-characteristics, theta-weighted diamond, weighted diamond + flux-fixup
 - Parallel first-collision
 - Analytic ray-tracing (DR)
 - Monte Carlo (DR and DD)
 - Multiple quadratures
 - Level-symmetric
 - Generalized Legendre Product
 - Galerkin



Denovo Parallel S_N

- Modern, Innovative, High-Performance Solvers
 - Within-group solvers
 - Krylov (GMRES, BiCGStab) and source iteration
 - DSA preconditioning (SuperLU/ML-preconditioned CG/PCG)
 - Multigroup solvers
 - Transport Two-Grid upscatter acceleration of Gauss-Seidel
 - Krylov (GMRES, BiCGtab)
 - Eigenvalue solvers
 - Power iteration (with rebalance)
 - CMFD in testing phase
 - Krylov (Arnoldi)
 - Shifted-inverse iteration in development

Power distribution in a BWR assembly

Denovo Capabilities

- **Parallel Algorithms**

- Koch-Baker-Alcouffe (KBA) wavefront decomposition
- Domain-replicated (DR) and domain-decomposed first-collision solvers
- Multilevel energy decomposition in development
- Parallel I/O built on SILO/HDF5

> 5M CPU hours on Jaguar with 2 bugs



2010 INCITE Award

Uncertainty Quantification for Three Dimensional Reactor Assembly Simulations, 8 MCPU-HOURS

2010 ASCR Joule Code

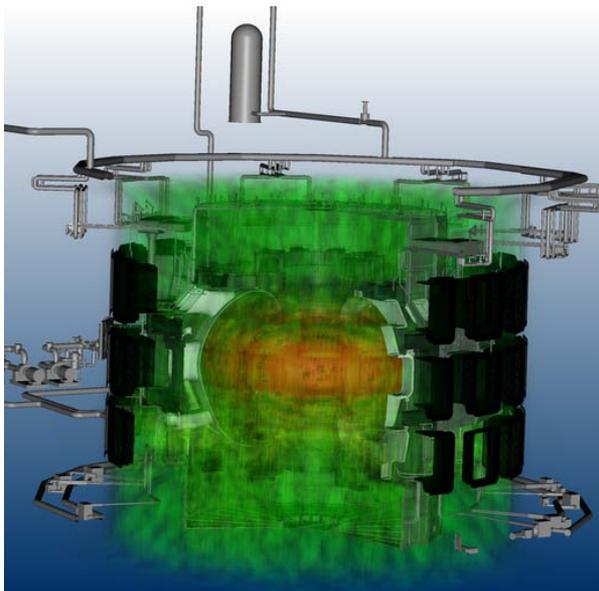
2009-2011 2 ORNL LDRDs

- **Advanced visualization, run-time, and development environment**

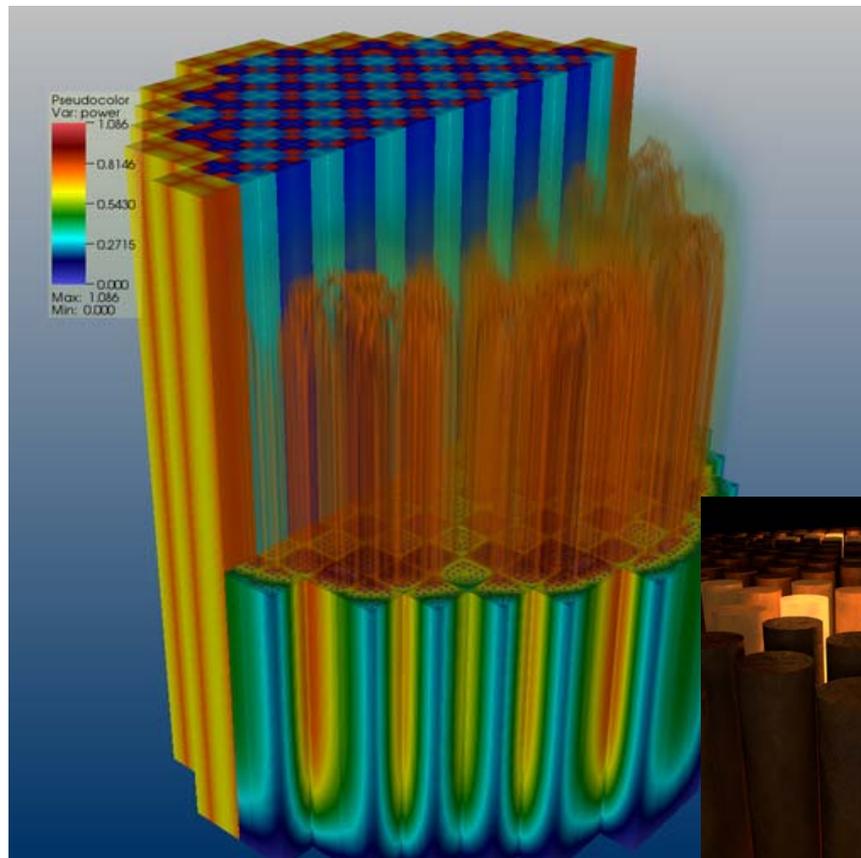
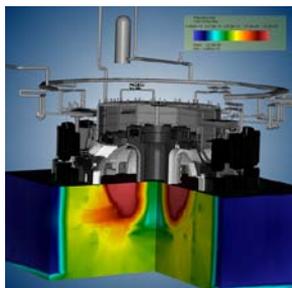
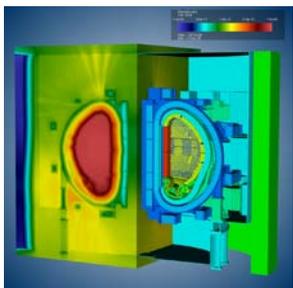
- 3 front-ends (HPC, SCALE, Python-bindings)
- Direct connection to SCALE geometry and data
- Direct connection to MCNP input through ADVANTG
- HDF5 output directly interfaced with VisIt
- Built-in unit-testing and regression harness with DBC
- Emacs-based code-development environment
- Support for multiple external vendors
 - GSL, BLAS/LAPACK, TRILINOS (required)
 - BRLCAD, SUPERLU/METIS, SILO/HDF5 (optional)
 - MPI (toggle for parallel/serial builds)
 - SPRNG (required for MC module)
 - PAPI (optional instrumentation)

Parallel Visualization

Using VisIt, Denovo is able to make 3D visualizations and perform data analysis on massive data sets.



ITER component performance/shielding



PWR900 reactor core benchmark

Discrete Ordinates Methods

- Operator form of the transport equation,

$$\mathbf{L}\psi = Q$$

- The S_N method is a collocation method in angle.
 - Energy is discretized in groups.
 - Scattering is expanded in Spherical Harmonics.
 - Multiple spatial discretizations are used (DGFEM, Characteristics, Cell-Balance).

$$\mathbf{L}\psi = \mathbf{MS}\phi + Q$$

$$\phi = \mathbf{D}\psi$$

Traditional S_N Solution Methods

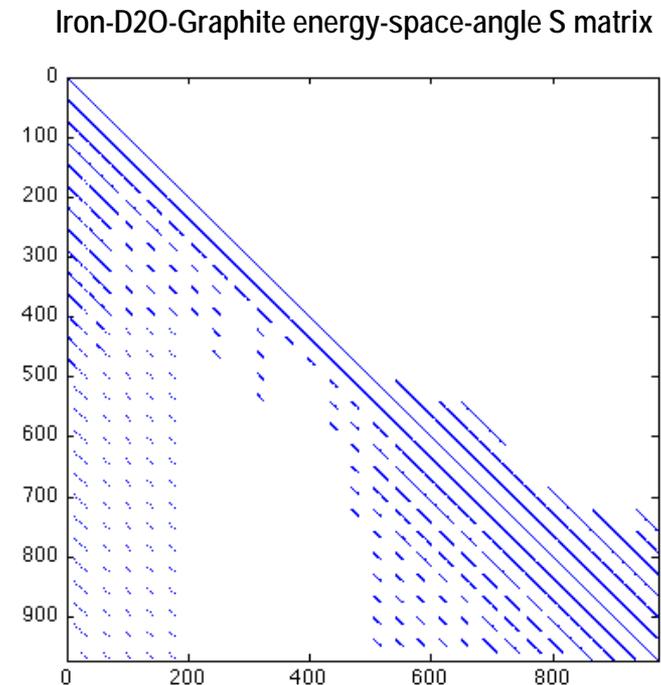
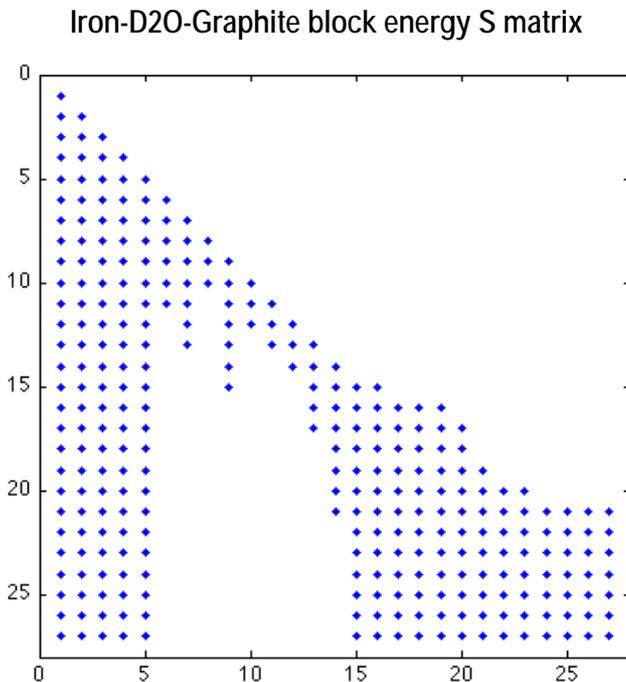
- Traditional S_N solutions are divided into **outer** iterations over energy and **inner** iterations over space-angle.
- Generally, accelerated Gauss-Seidel or SOR is used for outer iterations.
- Eigenvalue forms of the equation are solved using *Power Iteration*
- In Denovo we are motivated to look at more advanced solvers
 - Improved robustness
 - Improved efficiency
 - Improved parallelism

Krylov Methods

- Krylov methods are more robust than stationary solvers
 - Uniformly stable (preconditioned and unpreconditioned)
 - Compress space in an integral sense
- More efficient
 - Source iteration spectral radius $\rho(0)\epsilon = \frac{\sigma_s}{\sigma}\epsilon$
 - Gauss-Seidel spectral radius $\rho(0)\epsilon = (\mathbf{T} - \mathbf{S}_D)^{-1}\mathbf{S}_U\epsilon$
- There is no coupling in Krylov methods
 - Gauss-Seidel imposes coupling between rows in the matrix
 - Krylov has no coupling; opportunities for enhanced parallelism

Physics Dictates the Solution

- The Gauss-Seidel spectral radius for uniform graphite is 0.9812 = slow convergence
- Systems that are block-dense in energy are sparse in energy-space-angle
- Ideal candidates for Krylov methods



Krylov Methods for Discrete Ordinates

Gauss-Seidel Iteration in energy

$$\mathbf{L}_g \psi_g^{k+1} = \mathbf{MS}_{gg} \phi_g^{k+1} + \sum_{g'=0}^{g-1} \mathbf{MS}_{gg'} \phi_{g'}^{k+1} + \sum_{g'=g+1}^G \mathbf{MS}_{gg'} \phi_{g'}^k + Q_g$$

reduces to a series of one-group solves
(within-group **inner** iterations)

$$\mathbf{L}_g \psi_g = \mathbf{MS}_{gg} \phi_g + \bar{Q}_g$$

↓
up and down-scatter rolled into source

inners have the general form

$$\mathbf{L}\psi = \mathbf{M}\mathbf{S}\phi + q$$

Reformulating the Problem

$$\phi^{n+1} = \mathbf{DL}^{-1}(\mathbf{MS}\phi^n + q)$$

operate by \mathbf{DL}^{-1} to get Source Iteration

$$x^{n+1} = (\mathbf{I} - \mathbf{A})x^n + b$$

which is really fixed-point (Richardson) iteration

$$\mathbf{I} - \mathbf{A} = \mathbf{DL}^{-1}\mathbf{MS}$$

iteration matrix for Source Iteration

$$(\mathbf{I} - \mathbf{DL}^{-1}\mathbf{MS})\phi = \mathbf{DL}^{-1}q$$

put in form $\mathbf{Ax} = b$, we can use non-stationary iterative methods (**Krylov subspace**) to solve this linear problem

The inversion of \mathbf{L} is done using a wavefront solver that is implemented by solving for ϕ in the direction of particle flow → **Transport Sweep**.

Schur Complement

The S_N equations can be written as a 2 2 block-diagonal system,

$$\begin{pmatrix} \mathbf{L} & -\mathbf{MS} \\ -\mathbf{D} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \psi \\ \phi \end{pmatrix} = \begin{pmatrix} q \\ 0 \end{pmatrix}$$

The Schur Complement for block \mathbf{L} is

$$\mathbf{P} = \mathbf{I} - \mathbf{DL}^{-1}\mathbf{MS}$$

And the reduced space solution results

$$\begin{aligned} \mathbf{P}\phi &= g \\ g &= \mathbf{DL}^{-1}q \end{aligned}$$

Now we have a system where we invert a $(t \ t)$ and $(n \ n)$ system instead of one $(n+t) \ (n+t)$ system where t is dimensioned by the moments and n by angles.

Formulation of the Solution

1. Calculate the right-hand side (one transport sweep)

$$b = \mathbf{DL}^{-1}q$$

1. Apply the operator to the Krylov iteration vector until convergence (one transport sweep per Krylov iteration)

$$\mathbf{A}v^n \leftarrow y^n = (\mathbf{I} - \mathbf{DL}^{-1}\mathbf{MS})v^n$$

1. A pre-conditioner can also be applied

$$\mathbf{M}^{-1}\mathbf{A}v^n \leftarrow y^n = (\mathbf{I} + \mathbf{PC}^{-1}\mathbf{RS})(\mathbf{I} - \mathbf{DL}^{-1}\mathbf{MS})v^n$$

Krylov and the Multigroup Problem

- The full energy system can be solved by Krylov iteration as well

$$\left(\mathbf{I} - \begin{bmatrix} \mathbf{T}_0 & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{T}_G \end{bmatrix} \begin{bmatrix} \mathbf{M} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{00} & \dots & \mathbf{S}_{0G} \\ \vdots & \ddots & \vdots \\ \mathbf{S}_{G0} & \dots & \mathbf{S}_{GG} \end{bmatrix} \right) \begin{bmatrix} \phi_0 \\ \vdots \\ \phi_G \end{bmatrix} = \begin{bmatrix} \mathbf{T}_0 q_0 \\ \vdots \\ \mathbf{T}_G q_G \end{bmatrix}$$

- This has an identical form as the within-group equations with the exception that the iteration (Krylov) vector is now a function of energy

$$(\mathbf{I} - \mathbf{TMS})\phi = \mathbf{T}q$$

Eigenvalue Problem

- The eigenvalue problem has the following form

$$(\mathbf{I} - \mathbf{TMS})\phi = \frac{1}{k}\mathbf{TM}\chi\mathbf{f}^T\phi$$

- Expressed in standard form

$$\mathbf{A}x = kx$$

$$\mathbf{A} = (\mathbf{I} - \mathbf{TMS})^{-1}\mathbf{TM}\chi\mathbf{f}^T \quad x = \phi \quad \text{Energy-dependent}$$

$$\mathbf{A} = \mathbf{f}^T(\mathbf{I} - \mathbf{TMS})^{-1}\mathbf{TM}\chi \quad x = \mathbf{f}^T\phi \quad \text{Energy-independent}$$

- The traditional way to solve this problem is with *Power Iteration*

Advanced Eigenvalue Solvers

- We can use Krylov (Arnoldi) iteration to solve the eigenvalue problem more efficiently

$$y^k = \mathbf{A}v^k$$

Matrix-vector multiply and sweep $z^k = \mathbf{T}\mathbf{M}\chi\mathbf{f}^T v^k$

Multigroup fixed-source solve $(\mathbf{I} - \mathbf{T}\mathbf{M}\mathbf{S})y^k = z^k$

- Shifted-inverse iteration is also being developed (using Krylov to solve the shifted multigroup problem in each eigenvalue iteration)

$$(\mathbf{I} - \mathbf{T}\mathbf{M} \underbrace{(\mathbf{S} + \mu\mathbf{F})}_{\text{block-dense}})\phi = (\lambda - \mu)\mathbf{T}\mathbf{M}\mathbf{F}\phi$$

Residual Solvers

- In methods that require nested iteration (upscatter, eigenvalue, etc), Denovo solves the residual equation

$$\mathbf{P}\delta\phi = r_o$$

$$r_o = g - \mathbf{P}\phi_o$$

$$\delta\phi = \phi - \phi_o$$

- Then, the tolerance is relaxed

$$\epsilon' = \epsilon \frac{\|g\|}{\|r_o\|}$$

- Thus, as the outer iterations near convergence, the inner solutions require less work because the residual is within the specified tolerance

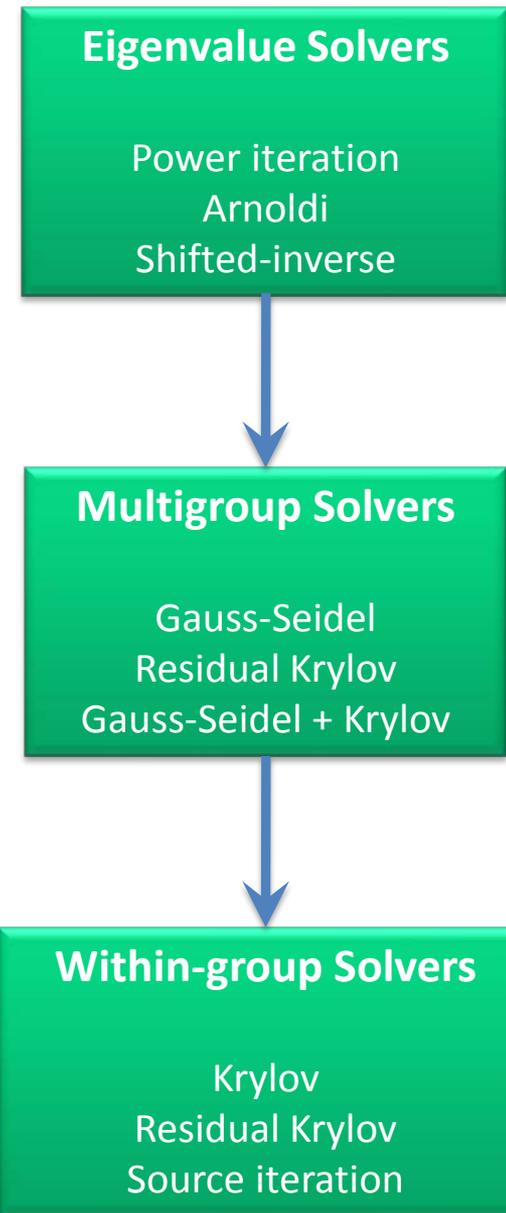
Solver Taxonomy

The innermost part of each solver are transport sweeps

$$y = \mathbf{T}z = \mathbf{D} \underbrace{\mathbf{L}^{-1}z}_{\mathbf{L}\psi = z}$$

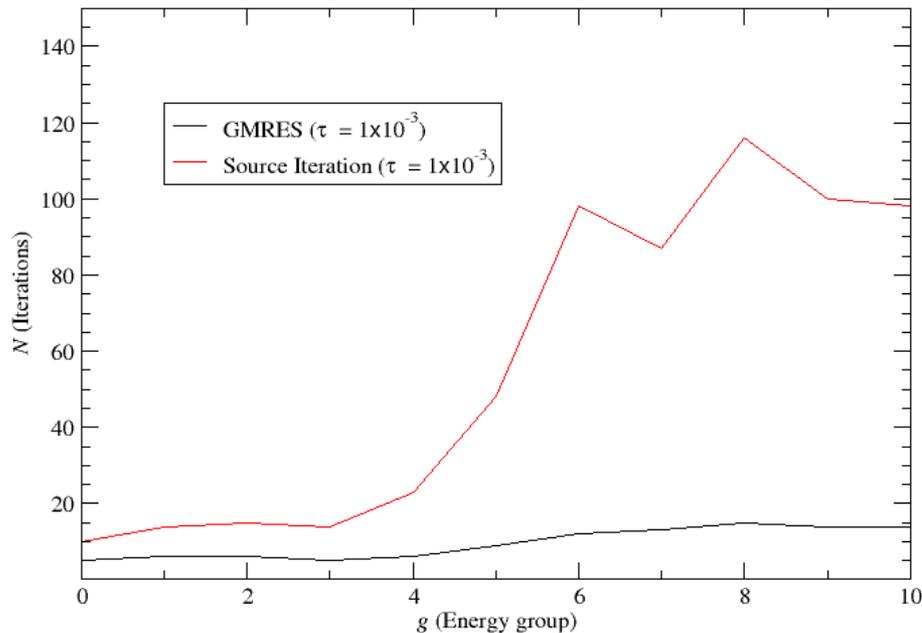


"It's turtles all the way down..."



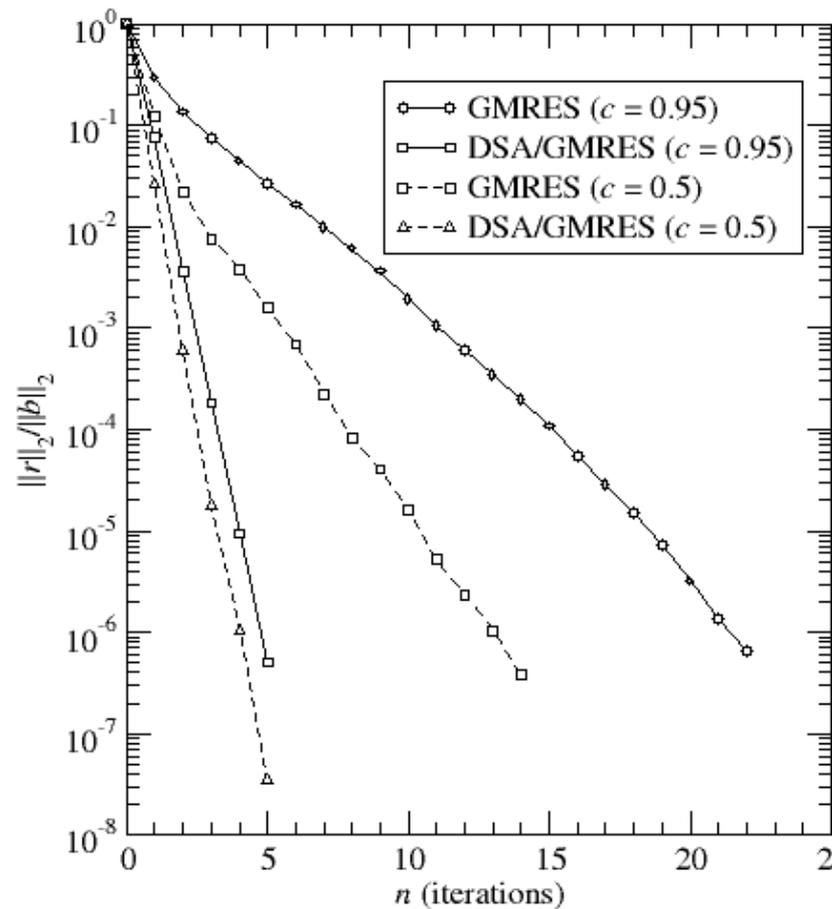
Advanced Solvers

- Regular GMRES performs very well without acceleration for most problems



- Transport two grid (TG) acceleration of upscatter is highly efficient and stable

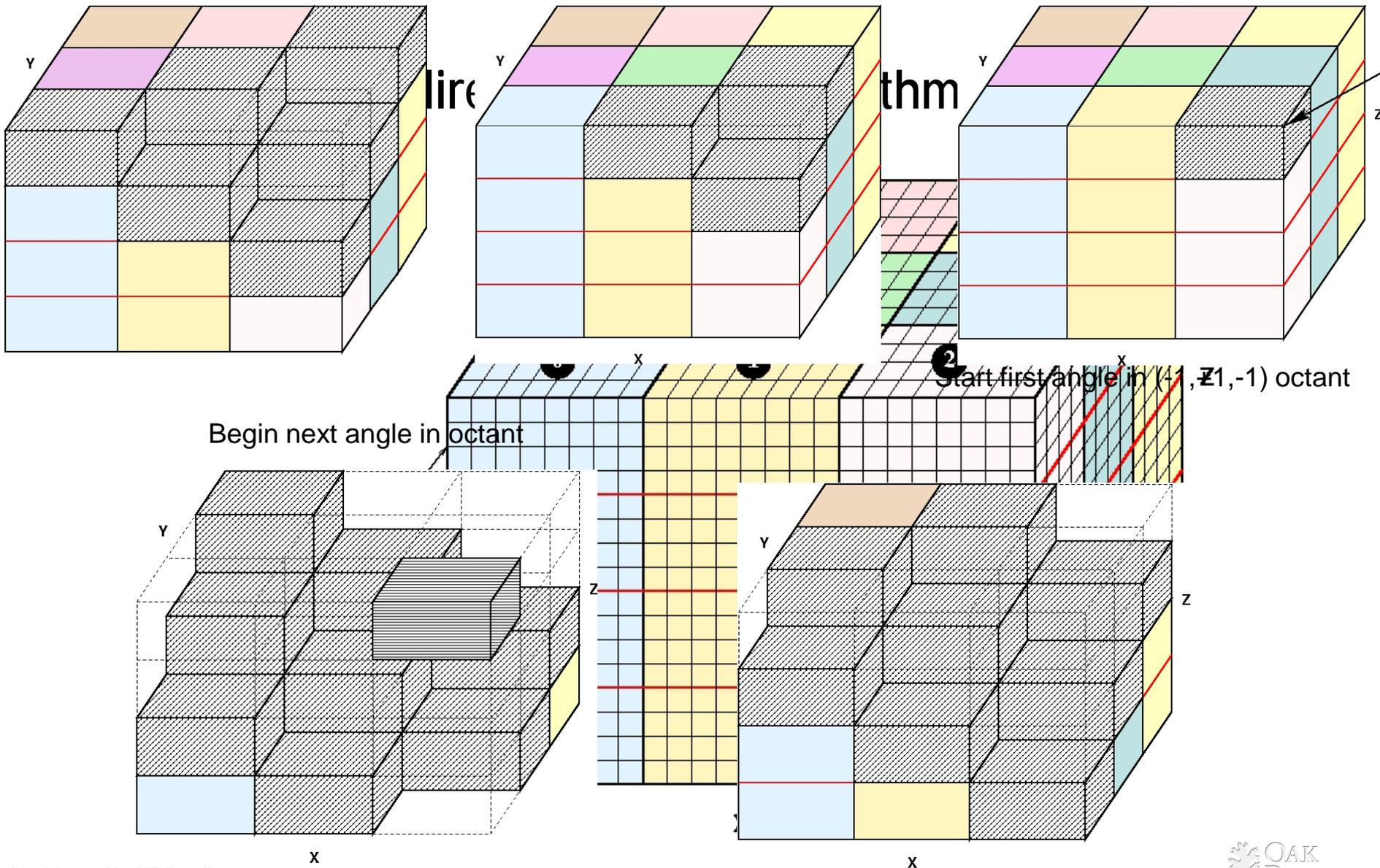
Method	Acc. S_N	GS Iterations	Normalized Time
Gauss-Seidel	-	175	1.000
Transport TG	8	15	0.113
Transport TG	4	14	0.097
Transport TG	2	13	0.086



- Preconditioning Krylov with DSA is unconditionally stable
- Excellent results for high scattering and/or tight convergence

KBA Algorithm

sweeping in direction of particle flow

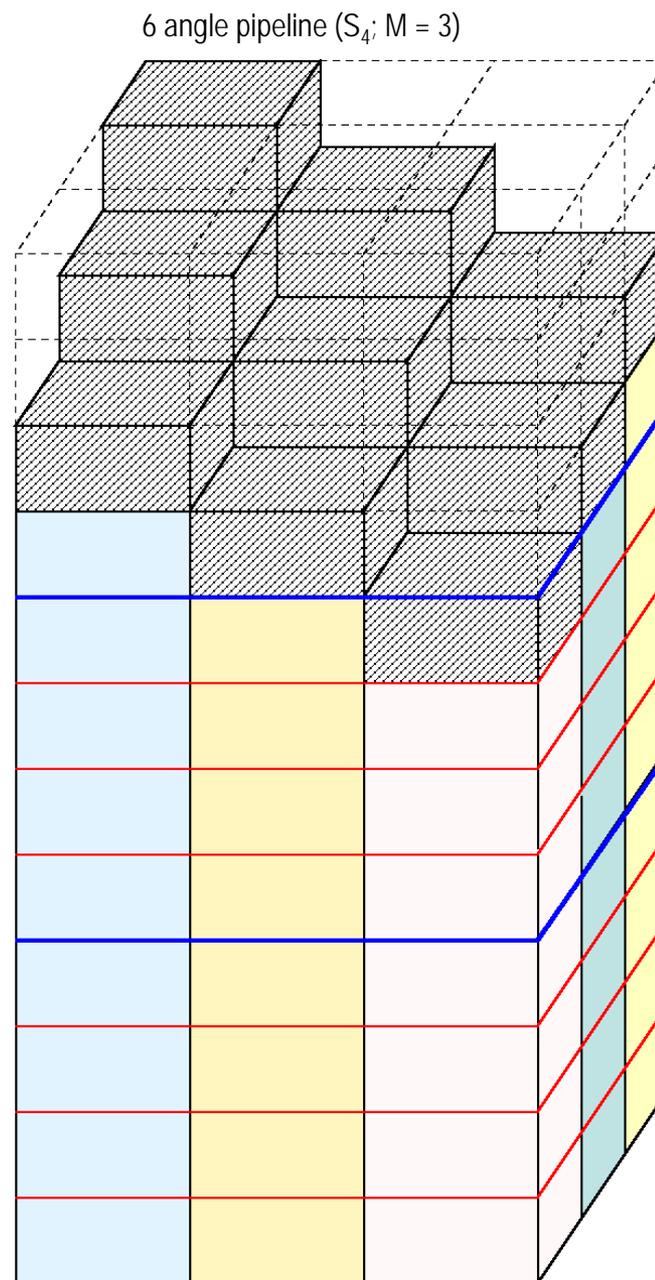


Parallel Performance

Angular Pipelining

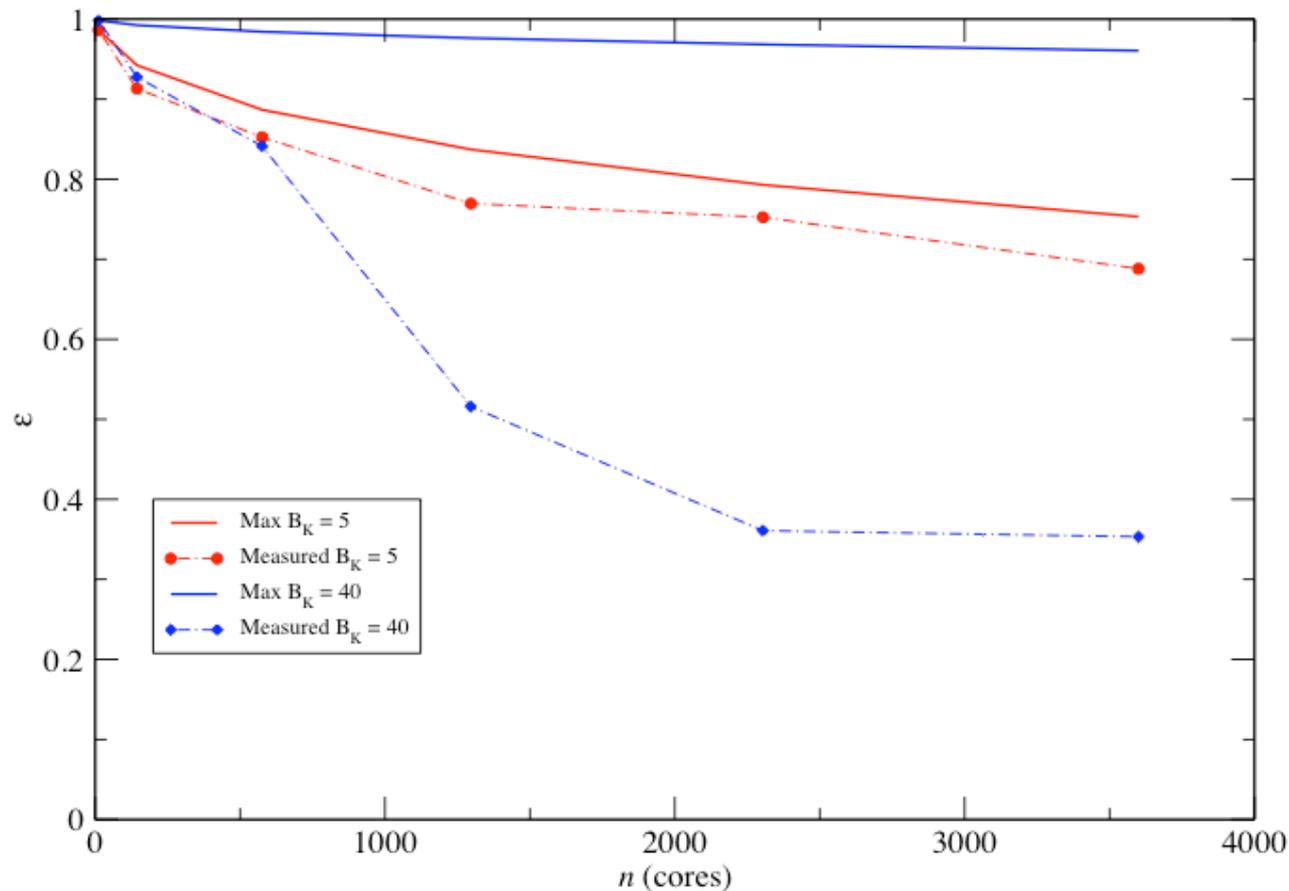
- Angles in $\pm z$ directions are pipelined
- Results in $2 \times M$ pipelined angles per octant
- Quadrants are ordered to reduce latency

$$\epsilon_{\max} = \frac{2MB_K}{2MB_K + P_I + P_J - 2}$$



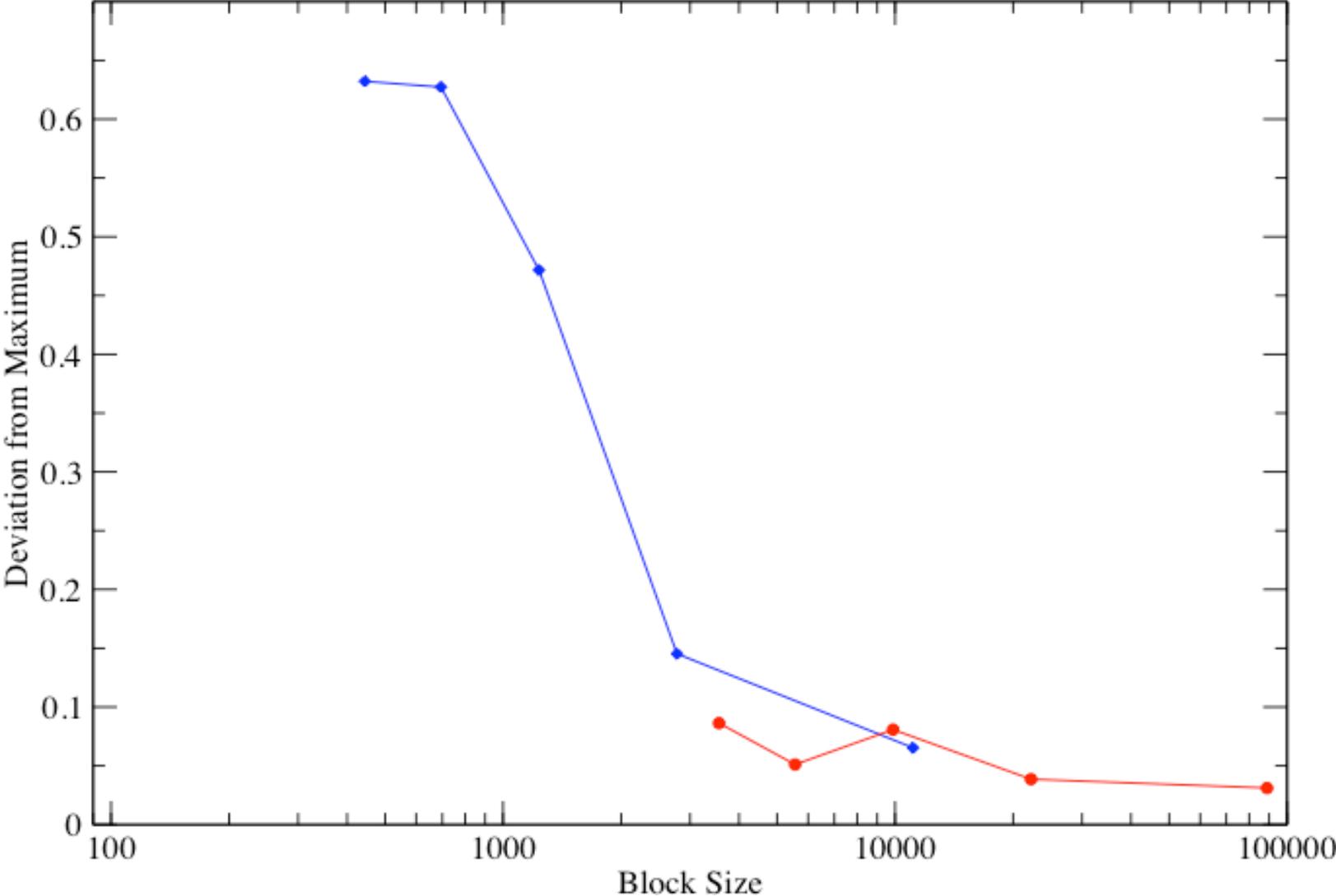
KBA Reality

KBA does not achieve close to the predicted maximum



- Communication latency dominates as the block size becomes small
- Using a larger block size helps achieve the predicted efficiency but,
 - Maximum achievable efficiency is lower
 - Places a fundamental limit on the number of cores that can be used for any given problem

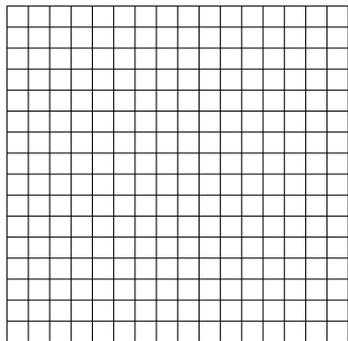
Efficiency vs Block Size



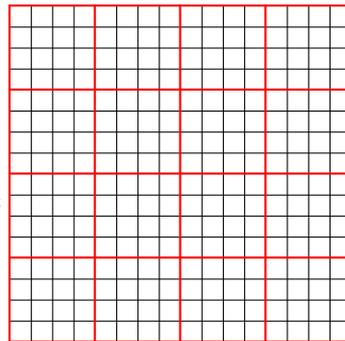
Overcoming Wavefront Challenge

- This behavior is systemic in any wavefront-type problem
 - Hyperbolic aspect of transport operator
- We need to exploit parallelism beyond space-angle
 - Energy
 - Time
- Amortize the inefficiency in KBA while still retaining direct inversion of the transport operator

Multilevel Energy Decomposition



4x4 *block* partitioning



3 *set* partitioning

12 12	13 13	14 14	15 15
8 8	9 9	10 10	11 11
4 4	5 5	6 6	7 7
0 0	1 1	2 2	3 3

0

12 28	13 29	14 30	15 31
8 24	9 25	10 26	11 27
4 20	5 21	6 22	7 23
0 16	1 17	2 18	3 19

1

12 44	13 45	14 46	15 47
8 40	9 41	10 42	11 43
4 36	5 37	6 38	7 39
0 32	1 33	2 34	3 35

2

0 Block id
0 Domain id
0 Set id

48 domains = 3 sets x 16 blocks

The use of Krylov methods to solve the multigroup equations effectively decouples energy

domains = blocks x sets
 $d = b + s \times \text{blocks}$

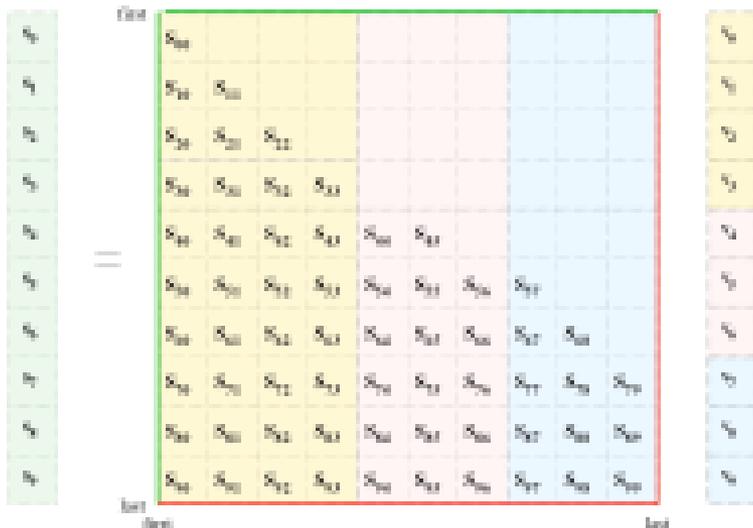
- Each energy-group S_N equation can be swept independently
- Efficiency is better than Gauss-Seidel

Sources for Transport Sweeps

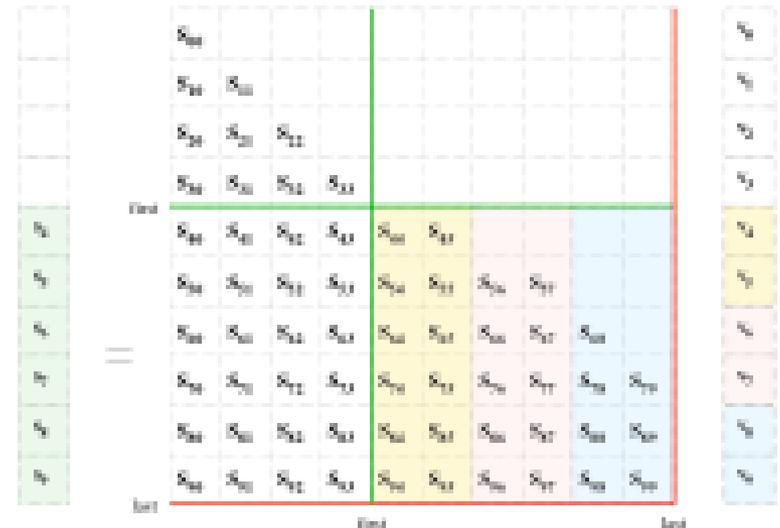
- In each energy set, the sweep source is coupled to all other sets

$$s_g = \mathbf{S}_{g0}v_0 + \mathbf{S}_{g1}v_1 + \cdots + \mathbf{S}_{gG}v_G$$

Decomposition over all groups



Decomposition over upscatter groups



Multilevel Summary

- Energy decomposed into sets.
- Each set contains blocks constituting the entire spatial mesh.
- The total number of domains is

$$\text{domains} = \text{sets} \times \text{blocks}$$

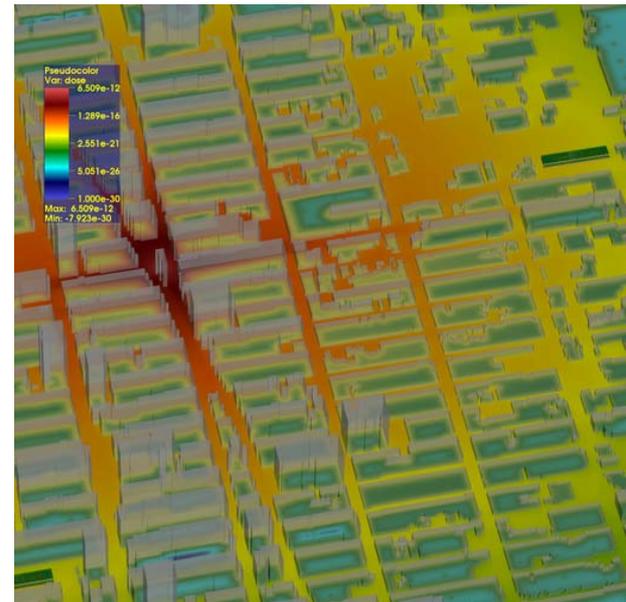
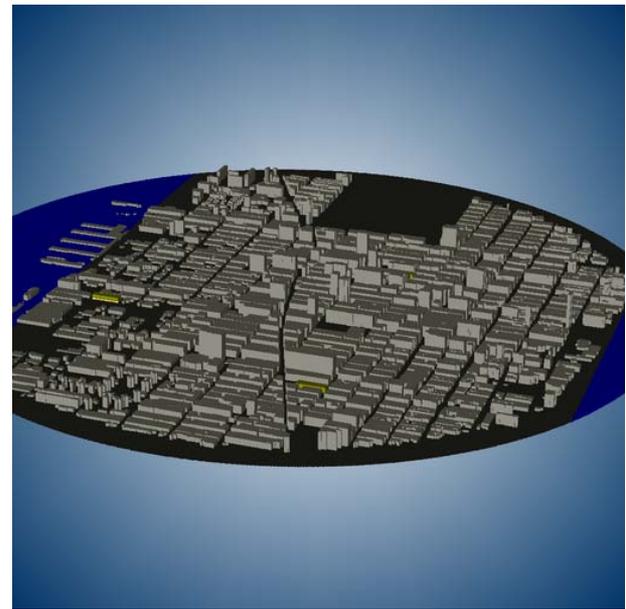
- KBA is performed for each group in a set across all of the blocks.
 - Not required to scale beyond $O(1000)$ cores.
- Scaling in energy across sets should be linear.
- Allows scaling to $O(100K)$ cores and enhanced parallelism on accelerators.

An Example

- We have seen good performance on 15,000 cores with
 - 250M cells
 - S_{16}
 - P_3
- To utilize the whole machine for a 44 group reactor calculation
 - 10 sets of 4 groups each (+4 additional groups in first 4 sets)
 - problem can now utilize 150,000 cores

Fixed-Source Problems

- Running very large urban models for national defense applications
 - 1.35×10^{14} total unknowns
 - Point-source located in center of model
 - Large streaming/near-void regions and ducts
- Ray-effect mitigation is **required**
 - First-collision source
 - Too many cells to replicate
 - Too many cores to efficiently use domain decomposition
 - Problem scale makes implementation difficult

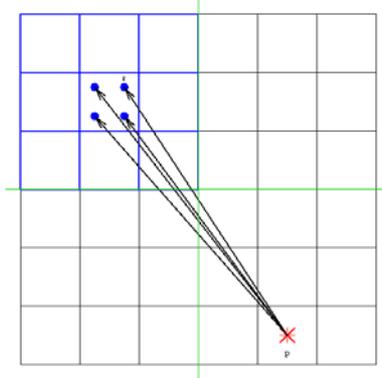


First-Collision Methods

- Denovo provides 2 first-collision methods for alleviating ray-effects

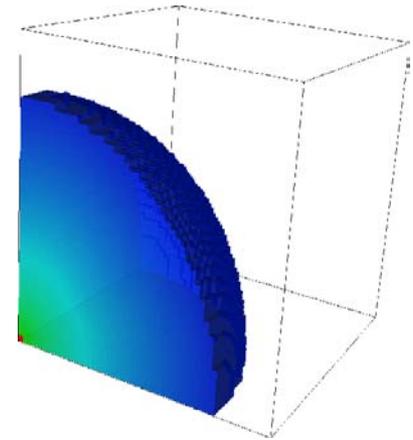
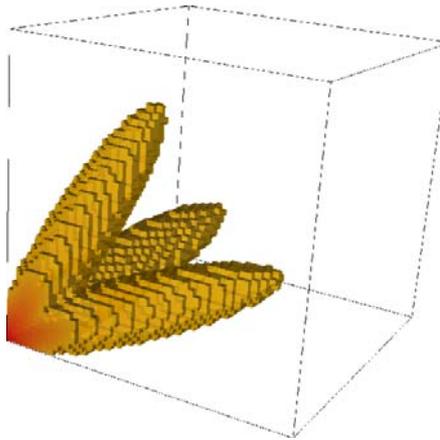
- Analytic ray-tracing

- Point sources
- Angular dependence
- Domain replication

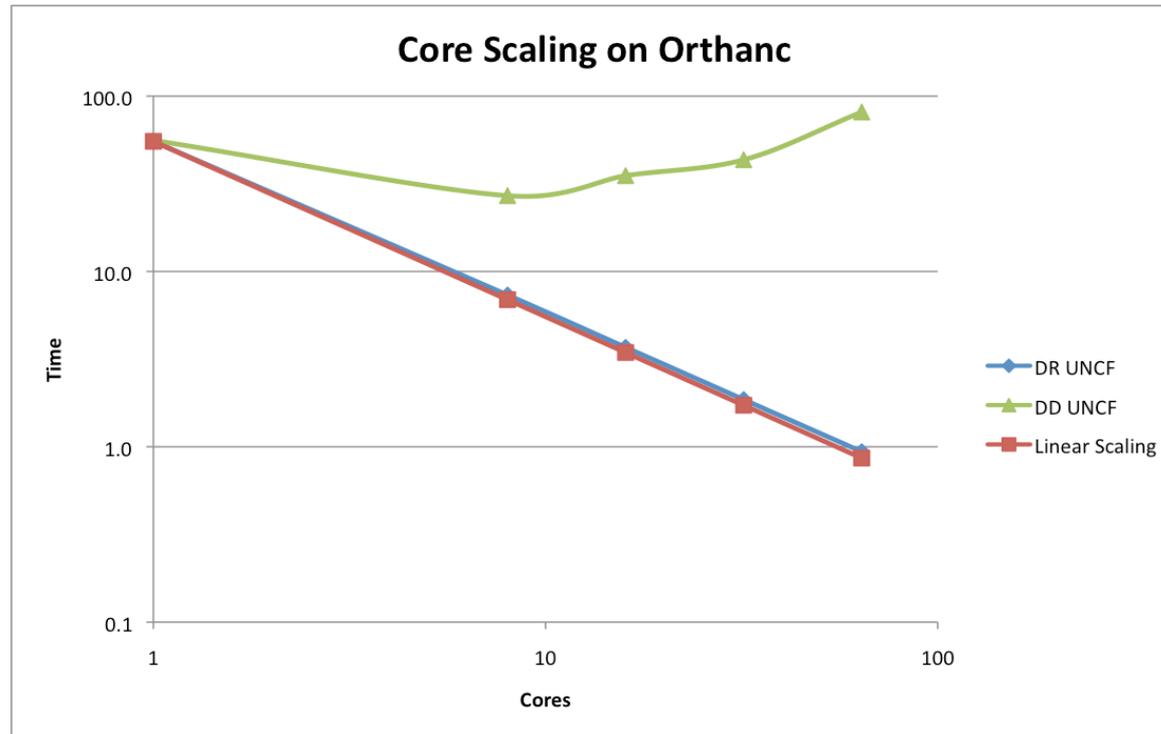


- Monte Carlo

- Point and distributed sources
- Angular dependence
- Domain replication
- Domain decomposition



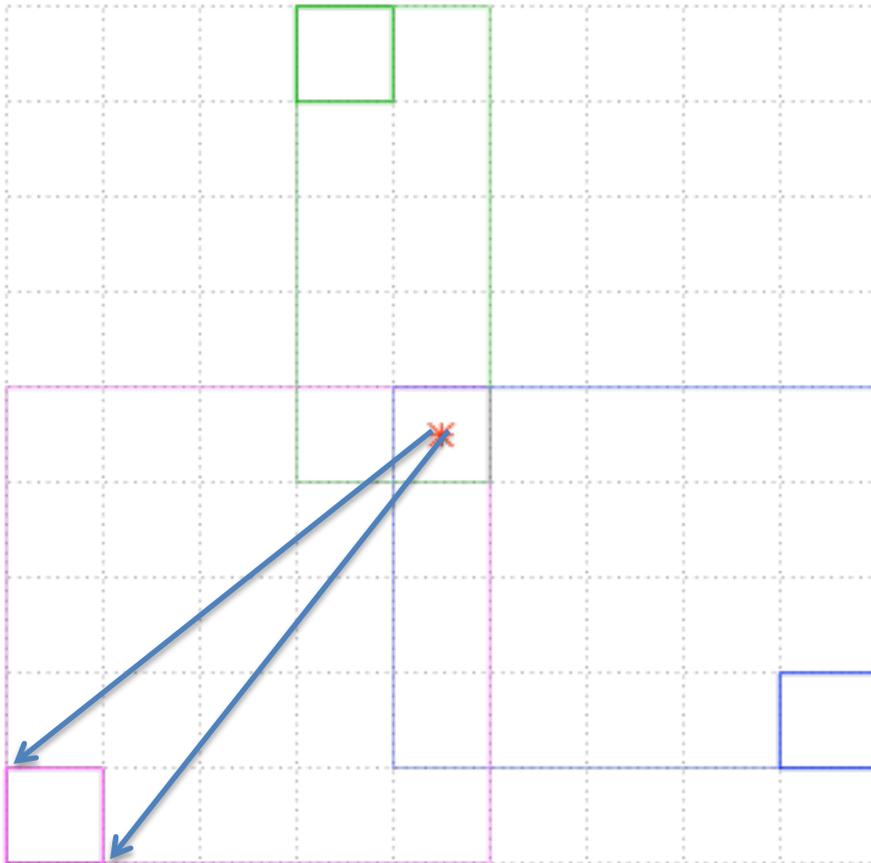
Monte Carlo First Collision Source



- As expected, DR Monte Carlo provides linear scaling
- DD scaling using the Urbatsch-Evans Milagro asynchronous transport algorithm still suffers from bad load imbalance
- We are looking at two methods for improving uncollided flux calculations
 - Reduced grid replication
 - Multilevel (in particles)

Improved Parallel First-Collision Methods

Use a reduced, replicated grid that still preserves replication for each domain



- Maximum replicated grid is 5 5 instead of 9 9
- Saves ~3.25 memory in ideal case (centered source).
- Non-centered sources will not realize as large a savings.
- Simple to implement.
- Further savings could be obtained by moving to a non-orthogonal mesh.
 - More expensive ray-tracing.
 - More expensive gathers during setup.

Summary

- We are hopeful that multilevel energy-space-angle decompositions will allow scaling of SN solutions to $O(100K)$ cores and beyond
- Multilevel energy parallelism is allowing implementation of an optimized sweep kernel on GPUs as part of the OLCF-3 effort
- Reduced-grid domain replication is being implemented to allow the use of first-collision sources (Monte Carlo and ray-tracing) on massive urban models

Questions?