L2:RTM.P2.01 Milestone Report
3D SN Transport
Tom Evans
Oak Ridge National Laboratory
June 30, 2011
3D SN Transport

Thomas M. Evans

CASL MNM PI Meeting

7-MAR-2011

ORNL, Oak Ridge, TN
Contributors

**ORNL Staff**
- Greg Davidson
- Josh Jarrell
- Bob Grove
- Chris Baker
- Andrew Godfrey
- Kevin Clarno
- Douglas Peplow
- Scott Mosher

**CASL**
- Roger Pawloski
- Brian Adams

**Students and PostDocs**
- Rachel Slaybaugh (Wisconsin)
- Stuart Slattery (Wisconsin)
- Josh Hykes (North Carolina State)
- Todd Evans (North Carolina State)
- Cyrus Proctor (North Carolina State)

**OLCF (NCCS) Support**
- Dave Pugmire
- Sean Ahern
- Wayne Joubert

**Other Funding Support**
- INCITE/ASCR/NRC/NNSA
Outline

• Parallel Algorithms and Solvers
• Subcell Discretization Methods
• Verification and Validation
Parallel Algorithms and Solvers
Denovo Capabilities

- State of the art transport methods
  - 3D/2D, non-uniform, regular grid $S_N$
  - Multigroup energy, anisotropic $P_N$ 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
      - QR
      - Galerkin

- 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)
      - Multigrid preconditioning in development
  - Eigenvalue solvers
    - Power iteration (with rebalance)
      - CMFD in testing phase
    - Krylov (Arnoldi)
    - RQI (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
  - Parallel I/O built on SILO/HDF5

> 10M CPU hours on Jaguar with 3 bugs

- **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
  - BLAS/LAPACK, TRILINOS (required)
  - BRLCAD, SUPERLU/METIS, SILO/HDF5 (optional)
  - MPI (toggle for parallel/serial builds)
  - SPRNG (required for MC module)
  - PAPI (optional instrumentation)

2010-11 INCITE Award
*Uncertainty Quantification for Three Dimensional Reactor Assembly Simulations*, 26 MCPU-HOURS

2010 ASCR Joule Code
2009-2011 2 ORNL LDRDs
Eigenvalue Problem

• The eigenvalue problem has the following form

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

• Expressed in standard form

\[A x = k x\]

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

• 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

\[ y^k = \Lambda v^k \]

Matrix-vector multiply and sweep

\[ z^k = TM \chi f^T v^k \]

Multigroup fixed-source solve

\[ (I - TMS) y^k = z^k \]

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

\[ (I - TM (S + \mu F)) \phi = (\lambda - \mu) TM F \phi \]

block-dense
Solver Taxonomy

The innermost part of each solver are transport sweeps

\[ y = Tz = DL^{-1}z \]

\[ L\psi = z \]

“It’s turtles all the way down…”
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.

$$
\epsilon_{\text{max}} = \frac{2MB_K}{2MB_K + P_I + P_J - 2}
$$
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
The use of Krylov methods to solve the multigroup equations effectively decouples energy

- Each energy-group $S_N$ equation can be swept independently
- Efficiency is better than Gauss-Seidel
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.
Whole Core Reactor Problem

PWR-900 Whole Core Problem
• 2 and 44-group, homogenized fuel pins
• 2×2 spatial discretization per fuel pin
• 17×17 fuel pins per assembly
• 289 assemblies (157 fuel, 132 reflector) – high, med, low enrichments
• Space-angle unknowns:
  – 233,858,800 cells
  – 168 angles (1 moment)
  – 1 spatial unknown per cell
## Results

<table>
<thead>
<tr>
<th>Solvers</th>
<th>Blocks</th>
<th>Sets</th>
<th>Domains</th>
<th>Solver Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI + MG GS (2-grid preconditioning)</td>
<td>17,424</td>
<td>1</td>
<td>17,424</td>
<td>11.00</td>
</tr>
<tr>
<td>PI + MG Krylov</td>
<td>10,200</td>
<td>2</td>
<td>20,400</td>
<td>3.03</td>
</tr>
<tr>
<td>Arnoldi + MG Krylov</td>
<td>10,200</td>
<td>2</td>
<td>20,400</td>
<td>2.05</td>
</tr>
</tbody>
</table>

Total unknowns = 78,576,556,800  
Number of groups = 2  
k_{eff} tolerance = 1.0e-3

- Arnoldi performs best, but is even more efficient at tighter convergence  
  - 27 v 127 iterations for eigenvector convergence of 0.001
- The GS solver cannot use more computational resource for a problem of this spatial size  
  - Simply using more spatial partitions will not result in more efficiency  
  - Problem cannot effectively use more cores to run a higher fidelity problem in energy
Parallel Scaling

17,424 cores is effectively the maximum that can be used by KBA alone.

Weak Scaling

- 1,728,684,249,600 unknowns (44 groups)
- 78,576,556,800 unknowns (2 groups)

Multilevel solvers allow weak scaling beyond the KBA wavefront limit.

MG Krylov solver partitioned across 11 sets.
Parallel Scaling and Peak Performance

17,424 cores is effectively the maximum that can be used by KBA alone.

Multilevel solvers allow weak scaling beyond the KBA wavefront limit.

1,728,684,249,600 unknowns (44 groups)

78,576,556,800 unknowns (2 groups)

Multilevel solvers allow weak scaling beyond the KBA wavefront limit.

MG Krylov solver partitioned across 11 sets.

CASL-U-2011-0018-000
Strong Scaling

Optimized communication gave performance boost to 100K core job, number of sets = 11

- Communication improvements were significant at 100K core level (using 11 sets).
- They do not appear to scale to 200K core. Why?
  - Multiset reduction each iteration imposes a constant cost!
Scaling Limitations

- Reduction across groups each iteration imposes a “flat” cost
- Only way to reduce this cost is to increase the work per set each iteration (more angles)
  - Generally the work in space will not increase because we attempt to keep the number of blocks per domain constant
RQI Solver

• Shift the right-hand side and take Rayleigh-Quotient accelerates eigenvalue convergence

\[(I - TMS\bar{S})\phi = (\lambda - \mu)T MF\phi\]

• In each inner we have the following multigroup problem to solve for the next eigenvector iterate, \(w \leftarrow w^{k+1}\)

\[(I - TMS\bar{S})w = b\]

• As this converges the MG problem becomes very difficult to solve (preconditioning is essential):

\[(I - TMS\bar{S})G^{-1}y = b \quad y = Gw\]
MG Krylov Preconditioning

- Each MG Krylov iteration involves two-steps
  - preconditioning: \( \mathbf{G} z^k = v^k \)
  - matrix-vector multiply: \( v^{k+1} = (\mathbf{I} - \mathbf{TMS}) z^k \)

- At end of iteration we must apply the preconditioner one last time to recover \( w^{k+1} \)

- We use a simple 1-D multigrid preconditioner in energy:
  \[ z \leftarrow \bar{\mathbf{G}} (z^{2h}, v^{2h}) \]
  - 1-pass V-cycle
V-Cycle Relaxation

- We are investigating both weighted-Jacobi

$$(I - TMS_D)z^{n+1} = TM(wS - S_D)z^n + wv + (1 - w)z^n$$

- And weighted-Richardson relaxation schemes

$$z^{n+1} = wTMS_z^n + wv + (1 - w)z^n$$

- Energy-parallelism is largely preserved
GPU Performance

- Single core (AMD Istanbul) / single GPU (Fermi C2050 comparison
- For both processors, code attains 10% of peak flop rate

<table>
<thead>
<tr>
<th></th>
<th>AMD Istanbul 1 core</th>
<th>NVIDIA C2050 Fermi</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel compute time</td>
<td>171 sec</td>
<td>3.2 sec</td>
<td>54X</td>
</tr>
<tr>
<td>PCIe-2 time (faces)</td>
<td>--</td>
<td>1.1 sec</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>171 sec</td>
<td>4.2 sec</td>
<td>40X</td>
</tr>
</tbody>
</table>
Subcell Discretization Methods
Subcell Interface Methodology

- Sub-divide orthogonal cell into two non-orthogonal “sub-cells” if two materials are in cell

- Uses Slice Balance Approach (SBA) to solve each sub-cell (initially step-characteristics)

- Conserves KBA sweep ordering
Current Efforts

• New Equation Type
  – Developed in Denovo’s DBC framework
  – SBA class templated on a slice equation (SC)

• New Mesh Type
  – Each cell must know its “cut” if it has one
  – Looking at automating this process using a volume-based algorithm (Young’s Method ($O(1)$) Level-sets, etc.)

• New Material Definitions
  – Must allow multiple materials in a cell
  – Still need to update fission sources to properly account for arbitrary “cuts” in a cell
Progress

• Initial test problem is an infinite medium, one material, four group eigenvalue problem.

• \( P_0 \) scattering, all reflecting boundaries, \( S_2 \) quadrature

• \( k_{\text{eff}} = 2.0, \frac{\Phi_0}{\Phi_1} = 5/3, \frac{\Phi_1}{\Phi_2} = 3/2, \frac{\Phi_2}{\Phi_3} = 4/5 \)

• Cut Cell Equations based on SC yield correct solution
Verification and Validation
Verification and Validation

• We have successfully run the C5G7 (unrodded) 3D and 2D benchmarks
  – All results within ~30 pcm of published benchmark
  – Linear-discontinuous spatial differencing (although SC differencing gave similar results)
  – Clean and mixed-cell material treatments (preserving absolute fissionable fuel volumes)
AMA V&V Activities

- Andrew Godfrey (AMA) has performed a systematic study of Denovo on a series of problems
  - 2/3D pins
  - 3x3 lattice
  - 17x17 full lattice
  - ¼ core

- Examined differencing schemes, quadratures, and solvers

- Of primary interest was the spatial resolution needed to obtain convergence (used Denovo python pincell-toolkit to generate meshes)

- Results compared to KENO runs using identical data
Pin Cell and Lattice Results

• Summary:
  – Pin cell yields converged results with 6x6 radial mesh and QR 4/4 quadrature (32 PCM agreement with 49 Groups)
  – Lattice yields converged results with 4x4 radial mesh and QR 2/4 quadrature (2 PCM agreement with 49 Groups); excellent pin-power agreement (< 0.3%)
  – Both problems converge consistently to 50x50 radial mesh and QR 10/10 quadrature
Quarter Core Simulations

- Radial zoning varied from 12x12 per fuel pin to 4x4.
- Axial zoning varied from 15.24 cm to 0.2 cm.

Conclusions:
- LD and TLD perform much better than SC when using coarse axial zoning.
- Good results are achieved (< 40 pcm) with LD using 4x4 radial zoning, 15.24 cm axial zoning, and QR 2/4 quadrature → results attained in 42 min runtime using 960 cores
  - 6x6 with QR 6/6 achieved best results at 15.24 cm axial (9 PCM)
- At 2.54 cm LD is still sufficiently more accurate than SC
- SC requires ~1 cm axial zoning to approach LD results
- Running a 1.6G-cell problem is feasible (190KCPU-hours)
Visualization

- Using VisIt’s parallel visualization and expression engine allows detailed analysis of these models